

國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

建構於 MPEG-4 之強韌性細粒狀可調層次式

視訊編碼技術之參數設定研究

A Study on Parameter Selection of Robust Fine
Granularity Scalable Video Coding Over MPEG-4

研究生 : 王紹宇

指導教授 : 王聖智 博士

中華民國 九十四年 九月

建構於 **MPEG-4** 之強韌性細粒狀可
調層次式視訊編碼技術之參數設定
研究

A Study on Parameter Selection
of Robust Fine Granularity
Scalable Video Coding Over
MPEG-4

研究生：王紹宇

Student：Shao-Yu Wang

指導教授：王聖智 博士

Advisor：Dr. Sheng-Jyh Wang



電子工程學系 電子研究所碩士班

碩士論文

A Thesis

Submitted to Institute of Electronics

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electronics Engineering

October 2005

Hsinchu, Taiwan, Republic of China.

建構於 MPEG-4 之強韌性細粒狀可 調層次式視訊編碼技術之參數設定 研究

研究生：王紹宇

指導教授：王聖智 博士

國立交通大學

電子工程學系 電子研究所碩士班



摘要

在這篇論文裡，以 RFGS 的架構作為基礎，我們提出一種依據影像特質，針對一段給定的頻寬，快速計算視訊編碼之 α 與 β 參數之最佳設定方法。在論文中，將對先前以 PSNR 為比較基礎的參數最佳選擇法作改善，改成以 MSE 為比較基礎，並以此為我們提出之快速參數選擇法的目標。首先，我們分析了不同影像在運動補償後的 residual 特性以及其和最後編碼大小的關係，並發現其規律性；另外我們也歸納出參數的調整在不同的頻寬條件下，與影像品質之間的關聯性。最後依照先前整理出的各種規律性，先建立起在單一頻寬時的參數預測模型，再進一步發展成考量一段給定頻寬的參數最佳化方法。我們提出的方法是利用簡單的運算即可對於不同的影像進行適當的編碼，以畫面(FRAME)為基礎，產生最佳的參數組合。

A Study on Parameter Selection of Robust Fine Granularity Scalable Video Coding Over MPEG-4

Student : Shao-Yu Wang

Advisor : Dr. Sheng-Jyh Wang

Department of Electronics Engineering, Institute of Electronics

National Chiao Tung University



In this thesis, based on the RFGS coding scheme, we propose an approach to automatically calculate the optimized α and β parameters for videos of different properties. Starting from the previous PSNR-based work, we propose new MSE-based approaches to pursue better video quality. We first analyze and model the residuals of motion compensation with different parameter setting under various bit rates. We then investigate the relationship between error drift and available bandwidth, and the relationship between MSE gain and extra bandwidth. With those relationships and models, we proposed first an approach to automatically choose the sets of optimized parameters for a fixed bit rate. Then we proposed a solution for the case with a range of bit rates. Frame-based approaches are studied and these proposed MSE-based approaches require only light computational loads.

誌謝

首先
感謝我的指導教授
王聖智老師
不只在論文上，課業上給我指導
更使我在處事的態度方法上
獲益良多

感謝實驗室的學長姊
帶領我進行研究

感謝同學們
常常給予我的鼓勵和支持



感謝實驗室的學弟們
夜深時陪我在實驗室渡過

感謝家人
對於我畢業時程失控的體諒
用關懷代替責備

感謝橋牌青年隊隊員們
你們送我的抱枕
讓我在實驗室躺著進入無數次的夢鄉

目錄

第一章 簡介.....	1
第二章 先前的研究.....	2
2.1 MPEG4 視訊編碼	2
2.1.1 移動補償(Motion Compensation)	3
2.1.2 離散餘弦轉換 (Discrete-Cosine-Transform)	3
2.1.3 鋸齒狀掃描 (Zigzag Scan)	4
2.1.4 變換長度編碼 (Run-Level-Coding).....	4
2.1.5 位元平面編碼 (Bit Plane Coding).....	4
2.1.6 可調層次式 (Scalability).....	5
2.1.6.1 訊雜比可調層次式(SNR Scalability).....	5
2.1.6.2 時間解析度可調層次式(Temporal Scalability)	6
2.1.6.3 空間解析度可調層次式(Spatial Scalability)	7
2.2 FGS (FINE GRANULAR SCALABILITY).....	8
2.3 一些改善 FGS 的方法.....	13
2.3.1 TFGS (Temporal – SNR FGS).....	13
2.3.2 MCFGS (Motion Compensation FGS).....	15
2.3.3 PFGS (Progressive FGS).....	17
2.4 RFGS (ROBUST FGS).....	20
2.4.1 LEAKY PREDICTION.....	23
2.4.2 PARTIAL PREDICTION	24
2.4.3 實驗結果	25
第三章 RFGS 最佳化參數挑選.....	27
3.1 先前的 RFGS 最佳化參數挑選方法.....	27
3.1.1 問題的定義.....	27
3.1.2 FRAME_BASED 最佳參數.....	28
3.1.3 MB_BASED 最佳參數	28
3.1.4 三階段搜尋法(THREE-STEP-SEARCH).....	31
3.2 MSE 最佳參數	31
3.3 兩個方法的結果比較	33
3.4 總結.....	39
第四章 數據分析.....	41
4.1 MSE GAIN	42

4.1.1	MSE GAIN 的來源	43
4.1.2	MSE GAIN 分析	45
4.1.3	總結(規則一).....	46
4.2	MSE LOST	47
4.2.1	錯誤漂移(ERROR DRIFT)	47
4.2.2	可加性	48
4.2.3	指數型遞減性.....	50
4.2.4	MSE LOST	51
4.2.5	總結(規則二、三).....	54
4.3	RESIDUAL 的特性	54
4.3.1	RESIDUAL 平均值.....	55
4.3.2	RESIDUAL BYTE 數.....	56
4.3.3	總結(規則四).....	57
第五章 提出的方法		60
5.1	方法一.....	60
5.2	方法二.....	61
5.3	方法三.....	62
5.4	方法四.....	62
第六章 實驗結果		64
6.1	方法一.....	64
6.2	方法二.....	65
6.3	方法三.....	67
6.4	方法四.....	68
第七章 結論		71



圖目錄

圖 2.1	MPEG4 編碼流程圖	2
圖 2.2	移動補償示意圖	3
圖 2.3	鋸齒狀掃描	4
圖 2.4	位元平面編碼	5
圖 2.5	雜訊比可調層次式	6
圖 2.6	時間解析度可調層次式	6
圖 2.7	空間解析度可調層次式	7
圖 2.8	不同編碼方式之比較	8
圖 2.9	FGS 示意圖	9
圖 2.10	FGS 編碼端架構圖	9
圖 2.11	FGS 解碼端架構圖	10
圖 2.12	FGS 資料分配示意圖	10
圖 2.13	FGS 和 SNR SCALABILITY 的比較	11
圖 2.14	FGS 和 NON SCALABLE CODING 的比較	12
圖 2.15	TFGS 編碼端	13
圖 2.16	TFGS 解碼端	14
圖 2.17	MCFGS 示意圖	15
圖 2.18	MCFGS 決策方塊圖	16
圖 2.19	MCFGS 相較於 FGS 的 PSNR 改善圖	17
圖 2.20	PFGS 精神示意圖	18
圖 2.21	PFGS 編碼端架構圖	18
圖 2.22	PFGS AKIYO COASTGUARD FOREMAN 結果示意圖	19
圖 2.23	RFGS 想法示意圖	20
圖 2.24	RFGS 編碼端架構圖	21
圖 2.25	RFGS 解碼端架構圖	22
圖 2.26	RFGS 線性模型示意圖	24

圖 2.27	AKIYO 實驗結果.....	25
圖 2.28	FOREMAN 實驗結果.....	25
圖 2.29	COASTGUARD 實驗結果.....	26
圖 2.30	RFGS 和 PFGS 的結果比較	26
圖 3.1	TR 示意圖.....	29
圖 3.2	LG OD 示意圖.	29
圖 3.3	TEST VECTORS 示意圖.....	30
圖 3.4	TEST VECTORS 結果示意圖.....	30
圖 3.5	PSNR 和 MSE 關係示意圖	32
圖 3.6	不同 PSNR 值時的 PSNR 影響	32
圖 3.7	NEWS FRAME-BASED 結果	34
圖 3.8	FOREMAN FRAME-BASED 結果	35
圖 3.9	STEFAN FRAME-BASED 結果.....	36
圖 3.10	MSE 方法和 PSNR 方法最佳參數比較圖	37
圖 3.11	平均 MSE PSNR 結果比較圖	38
圖 3.12	最佳化流程示意圖	39
圖 4.1	MSE 估算示意圖	42
圖 4.2	MSE GAIN 來源分析	43
圖 4.3	RESIDUAL 大小示意圖.....	44
圖 4.4	RESIDUAL 大小差值示意圖.....	44
圖 4.5	MSE 和 BYTE 數關係圖.....	45
圖 4.6	MSE GAIN 和 MSE 值關係圖.....	46
圖 4.7	ERROR DRIFT 示意圖	48
圖 4.8	可加性實驗示意圖	49
圖 4.9	可加性實驗結果示意圖	50
圖 4.10	指數型遞減性實驗結果示意圖	51
圖 4.11	ERROR DRIFT 大小示意圖.....	52
圖 4.12	MSE LOST 和 BYTE 數關係圖(NEWS).....	52
圖 4.13	MSE LOST 和 BYTE 數關係圖(FOREMAN).....	53

圖 4.14	MSE LOST 和 BYTE 數關係圖(STEFAN)	53
圖 4.15	ENCODER 示意圖	55
圖 4.16	RESIDUAL 平均值示意圖	55
圖 4.17	RESIDUAL BYTE 數示意圖	57
圖 4.18	RESIDUAL 線性模型 (NEWS)	58
圖 4.19	RESIDUAL 線性模型 (NEWS FOREMAN STEFAN)	58
圖 4.20	編碼端參數決定示意圖	59
圖 5.1	方法一示意圖	60
圖 5.2	方法二示意圖	61
圖 5.3	方法三示意圖	62
圖 5.4	方法四示意圖	63
圖 6.1	方法一結果圖	65
圖 6.2	方法二結果圖	66
圖 6.3	方法三結果圖(NEWS)	67
圖 6.4	方法三結果圖(FOREMAN)	68
圖 6.5	方法四結果圖(NEWS)	69
圖 6.6	方法四結果圖(FOREMAN)	69
圖 6.7	方法四結果圖(STEFAN)	70

表目錄

表 2.1	TFGS 效能比較表.....	14
表 4.1	MSE GAIN 和 BYTE 數關係表	46
表 4.2	MSE LOST 和 BYTE 數關係表	54
表 4.3	RESIDUAL 平均值和 BYTE 數關係表	59



第一章 簡介

網路是最方便最普遍的影像傳播媒介，而隨著有線網路，無線網路的蓬勃發展，如何提供有效的壓縮方式，使得較佳品質、資料量較大的影像可以被順利地傳輸，已經變成了一個重要的議題。目前，在 MPEG-4 編碼標中提供了，細粒狀可調層次式編碼[Fine Granularity Scalable Video Coding]的功能，可以使得編碼結果適用於一整段的可能頻寬，也符合了實際的網路傳輸須求。而本論文將以改良過的 RFGS [Robust Fine Granularity Scalable Video Coding] 架構為基礎，對於不同影像特色進行討論，提出一個可以經過簡單計算出較佳參數組合的方法。

本論文的架構如下，第二章將介紹先前的研究，包含 MPEG-4 標準、FGS、其他 FGS 的改善方式，以及 RFGS。第三章將介紹先前學長以 PSNR 為考量基礎來計算影像最佳參數組合的方法，並提出以 MSE 為考量基礎的方式，來提升整體的影像品質。第四章將分析不同影像的 residual 經過編碼後的資料量大小和參數選擇之間的關係，並且建立起頻寬和 error drift 以及 MSE 大小的關係模型。在接下來的第五章中，則依照先前得到的規律和模型，提出幾個以畫面為單位的計算最佳參數組合的演算法。第六章將對於實驗的結果進行比較和討論。最後在第七章中，將提出一個結論。



接下來的幾個小節將對於 MPEG-4 編碼中比較重要特別的方法作簡單的介紹，分別是運動補償，離散餘弦轉換，鋸齒狀掃描，變換長度編碼和位元平面編碼。

2.1.1 運動補償(Motion Compensation)

運動補償是相當基本而重要的視訊編碼過程，可以達到大量的壓縮效果，由於一般視訊具有大量的 Temporal Redundancy，也就是接連的幾張畫面，甚至是一整串的畫面，彼此的關聯性都很高，經過運動補償，就是直接取用前面一張畫面的資料，只對於不同的部份進行編碼，這樣就可以大量的降低資料量，達到壓縮的目的。運動補償的過程簡述如下：

- 以區塊為基礎 (Block-based)
把畫面切割為許多小單位，通常為正方塊。
- 運動估測 (Motion Estimation)
對於每一個方塊，在前一張畫面中尋找最相似的方塊，然後記錄其位置並且同時記錄相對應的運動向量 (Motion Vector)，將位置和運動向量置放在最後編碼出來的 bit-stream 中
- 運動補償
把目前的方塊的影像數值減去運動估測對應的方塊影像數值，剩下的影像數值就是 Residual，分為 Y，U，V 三個部份

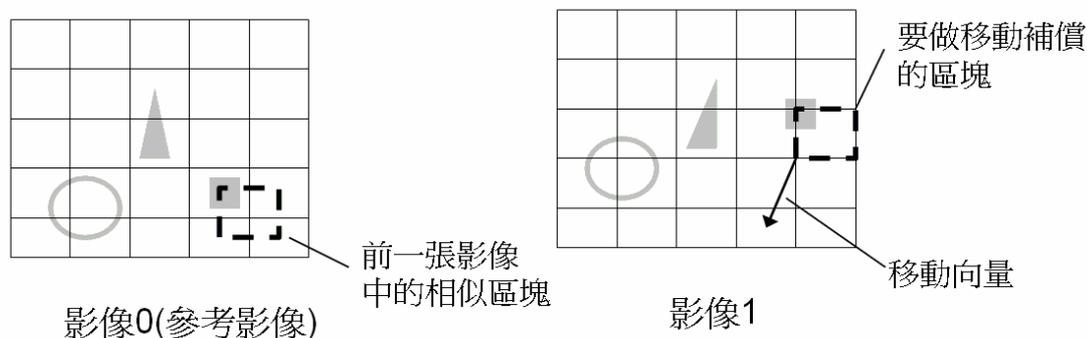


圖 2.2 運動補償示意圖[5]

2.1.2 離散餘弦轉換 (Discrete-Cosine-Transform)

離散餘弦轉換是影像壓縮與視訊壓縮常見的方法，在 MPEG-4 的標準中，就是採用 DCT 係數來壓縮，由於人眼對於高頻訊號比較不敏感，所以影像經過離散餘弦轉換之後，高頻的係數可以比較粗略地量化，如此一來，可以大幅增加壓縮比，而且對於影像品質的影響很小。

2.1.3 鋸齒狀掃描 (Zigzag Scan)

二維的離散餘弦轉換係數，經過量化後，高頻部分數值很小，幾乎皆為零，能量集中在低頻係數。而鋸齒狀掃描將係數由低頻排列至高頻，高頻的零被集中排列，這樣子的排列順序經由變動長度編碼(Run Level Coding)，可達到可觀的壓縮比。圖示如下：

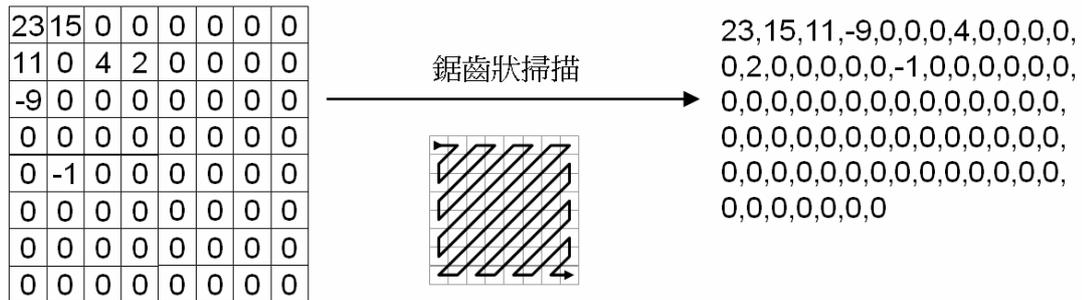


圖 2.3 鋸齒狀掃描[5]

2.1.4 變換長度編碼 (Run-Level-Coding)

變換長度編碼對於有很多一連串的零的序列，能有很好的壓縮效果，經過鋸齒狀掃描後，產生的序列，能量集中在前面，後面的數值大多是零，因此 MPEG-4 採用變換長度編碼對序列壓縮。壓縮鋸齒狀掃描後的序列，延續前一小節的例子，我們只要用 7 個變換長度編碼符號(Symbol)即可完成長度為 64 的序列的編碼，過程如下：

(Run, Level, End) → (0,23,0) (0,15,0) (0,11,0) (0,-9,0) (3,4,0) (5,2,0) (5,-1,1)

各個不同的符號，根據機率的統計，再施以霍夫曼編碼(Huffman coding)。

2.1.5 位元平面編碼 (Bit Plane Coding)

位元平面編碼是 MPEG-4 達成「可精緻可調層次式(Fine-Granularity-Scalability)」的主要步驟。離散餘弦轉換的係數，經過量化後，編碼成爲基本層(Base-layer)。而量化誤差則進一步經由位元平面編碼，編碼成加強層(Enhancement-layer)，可以擷取不同資料量解碼，還原出不同的影像品質。位元平面編碼的過程簡述如下：

- 把量化誤差以二進數表示，並且對齊最小有效位(LSB)
- 找出所有數值中的最高有效位元(MSB)

- 從最高有效位元開始，一層一層的對每個位元層編碼
- 每一個位元層，分別做鋸齒狀掃描與變換長度編碼

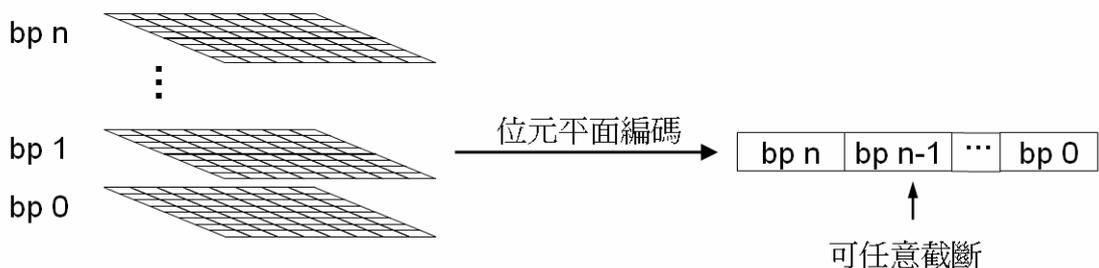


圖 2.4 位元平面編碼[5]

如此一來，接收端若是接收了全部的資料，則可以還原出完整的數值；接收端若是接收了一部分的資料，還是可以還原出所有的數值，只是比較不精確而已。

2.1.6 可調層次式(Scalability)

廣義來說，MPEG-4 的可調層次式有四種，雜訊比可調層次式、時間解析度可調層次式、空間解析度可調層次式、還有資料分割(Data Partitioning)。其中資料分割主要用於資料保護，不如前三者著重於精細可調層次式的實現。所以接下來我們主要分別簡述前三項。

2.1.6.1 訊雜比可調層次式(SNR Scalability)

這裡所謂的雜訊，並不是傳輸上的通道雜訊。一般編碼規格所謂的訊雜比，其雜訊(Noise)是指量化雜訊(Quantization Noise)，而 MPEG-4 可精緻可調層次式編碼(Fine-Granularity-Scalability Coding)主要方法是透過位元平面編碼達成，而位元平面編碼所送達的位元平面層數多寡，直接影響到視訊訊雜比，故訊雜比可調層次式經由位元平面編碼，即可達到相當精細的程度。

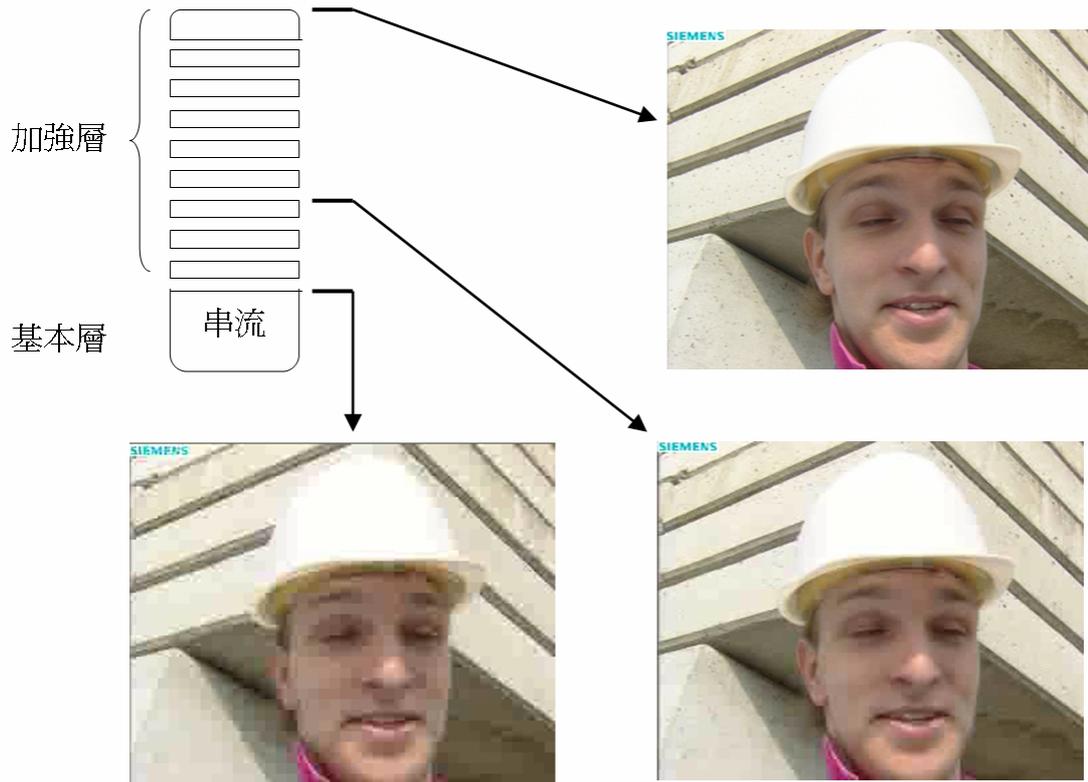


圖 2.5 雜訊比可調層次式[5]

2.1.6.2 時間解析度可調層次式(Temporal Scalability)

一般視訊的時間取樣頻率為每秒 24~30 張，而時間解析度可調層次式即是調整時間取樣頻率。例如：視訊資料編碼成爲兩個串流(stream)，只接收第一個串流可解碼還原出每秒 15 張畫面，如再接收第二個串流就可以還原出每秒 30 張畫面，而從 15 張畫面到 30 張畫面之間，也可以存在許多等級的時間解析度，端看編碼規格支援與否。



圖 2.6 時間解析度可調層次式[5]

2.1.6.3 空間解析度可調層次式(Spatial Scalability)

空間解析度可調層次式所調整的，是視訊檔案的解析度。例如：一個 CIF 格式的視訊檔案，其畫面解析度為 352 X 288，經過具有空間解析度可調層次式能力的編碼器編碼，形成兩個串流，只接收第一個串流可解碼還原出 QCIF 格式的視訊(176X144)，再接受第二個串流就可以還原出原來 CIF 格式的畫面。其可調層次式能力的精細程度，端看編碼規格支援與否。

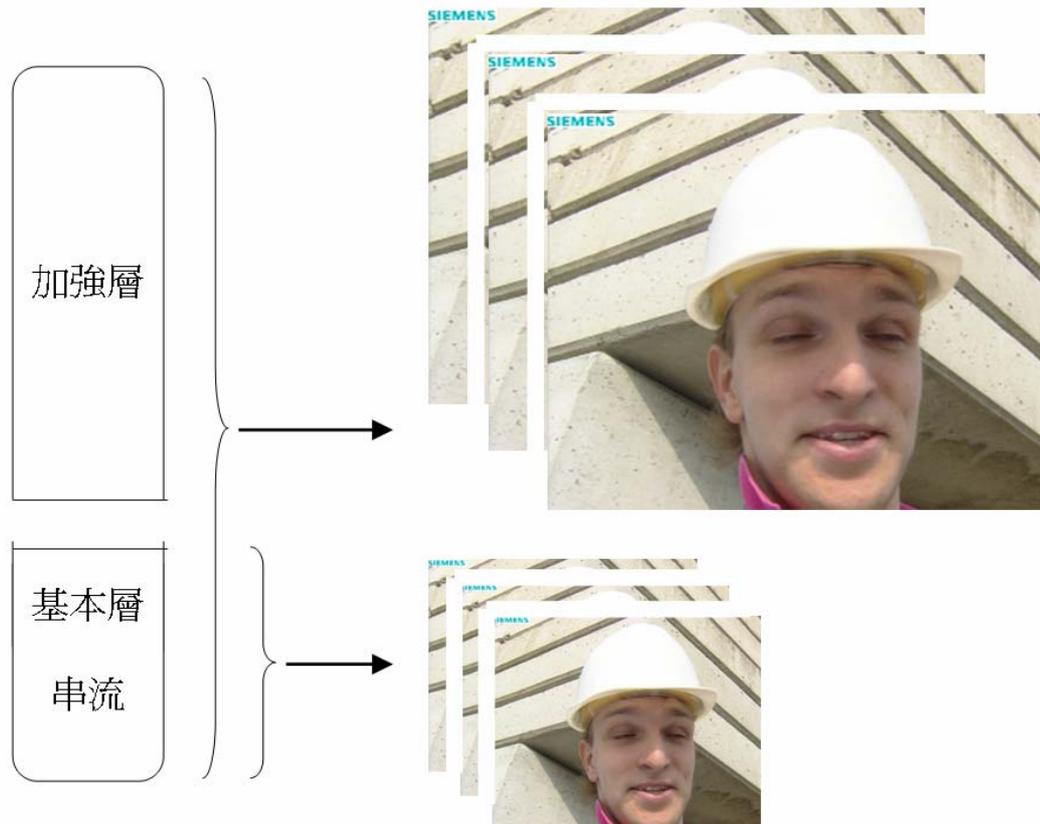


圖 2.7 空間解析度可調層次式[5]

一般來說，時間解析度可調層次式與空間解析度可調層次式的可調層次式程度，不及雜訊比可調層次式來得精細(Fine-Granularity)。一方面是因為三種可調層次式在本質上的難易度不同，另一方面因為目前這兩種可調層次式所採用的方法，其可調層次式能力皆不及雜訊比可調層次式所採用的位元平面編碼。

2.2 FGS (Fine Granular Scalability)

在先前的 MPEG 標準中，是在固定的頻寬下，對編碼作最佳化，但是在目前的影像傳輸中，因為使用網路來作為影像傳輸的媒介，而實際的網路環境，是會有變動的、不穩定的，為了在這樣的條件下，能得到比較好的影像品質，需要在編碼時，對一個區段的頻寬來作最佳化的動作，而不再只是對於單一的頻寬來作最佳化。有鑑於此，在 MPEG-4 標準中，加入了細粒狀可調層次式視訊編碼技術[*fine granularity scalable video coding*]這項技術，希望可以逼近於 *Distortion-Rate* 曲線的效能。下圖為使用不同的編碼方式所得到的影像品質和頻寬的示意圖，在本節接下來的篇幅中，我們將簡單的介紹 FGS 技術的方法和精神。

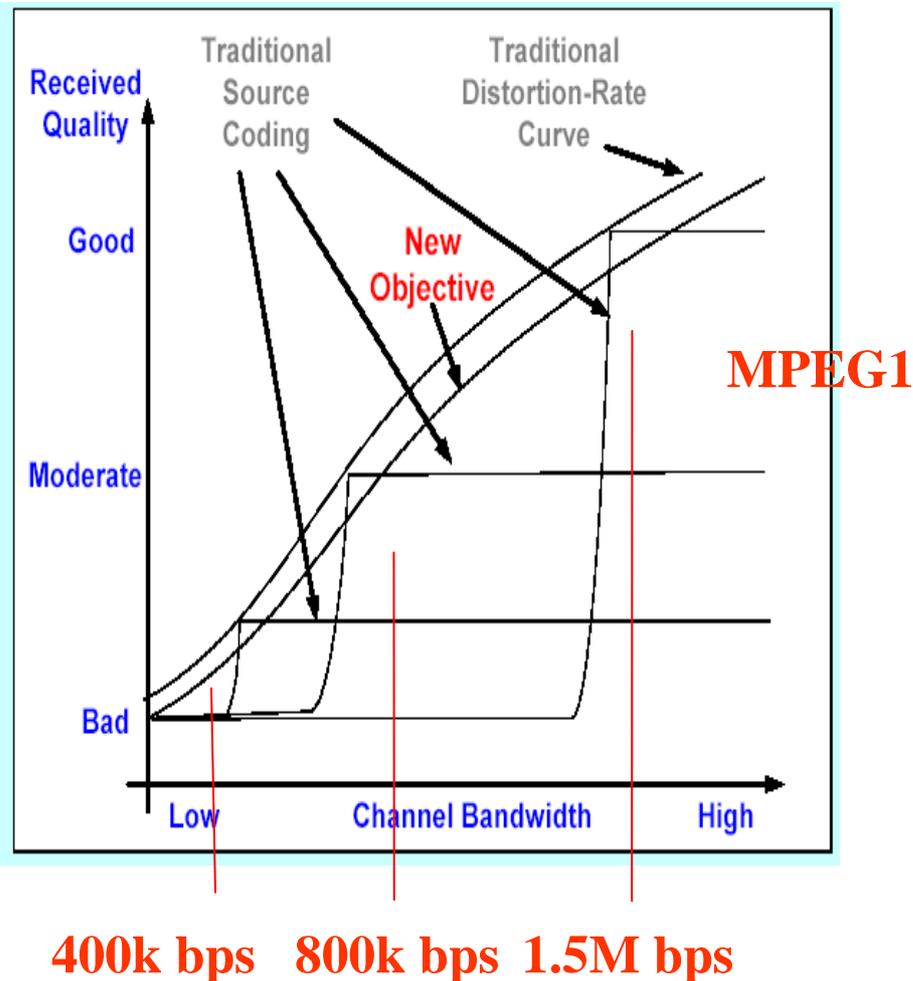


圖 2.8 不同編碼方式之比較

FGS 把影像的編碼分成兩個部份，基本層[Base Layer]和加強層[Enhancement Layer]。其中最重要的資料都是放在基本層的部份，這些資料量在最差的頻寬條件下都假設能夠被完整地傳輸，因此必須依照可能的頻寬分佈，對基本層的大小作調整。為了達到最好的壓

縮率，使得編出來的碼最小，所以本層的編碼方式是採用非可調層次式編碼[non scalable coding]。而加強層的部份，則是採用位元平面編碼[bit-plane coding]，將全部的資料量扣除掉基本層已包含的部份，按照量化後的大小，編碼成數個位元平面。依照當時的頻寬狀況，解碼端會得到不等量的加強層資料，但是不管有多少的資料因為網路狀況而被丟棄，解碼端都可以以基本層和部份的加強層資料，解出完整的影像，差別只是在於品質的好壞而已。採用這樣的編碼方式，可以使得影像經過編碼後，可以適用於一整個區段的頻寬，而不在只是單一的頻寬。下圖是示意圖：

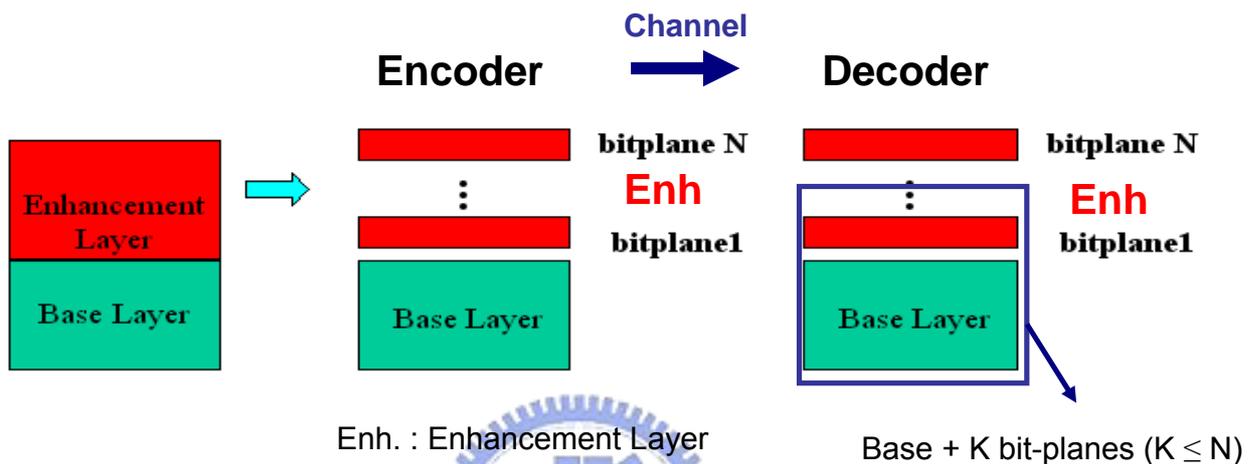


圖 2.9 FGS 示意圖

在 FGS 編碼中，除了有經過量化(quantization)之外，也將影像的數值轉換成 DCT 係數，希望藉由它高低頻分佈的特質，配合上 VLC 編碼，可以達到比較好的壓縮效果。上述的方法在之前已經已介紹過，這裡不再重複。而在解碼端的部份則有很多大同小異的方式可供選擇，大致上就是一段一段的讀入資料，直到發現下一張 frame 的 header 為止，下面的兩張示意圖，分別表示出編碼端(encoder)和解碼端(decoder)的一些細節部份。

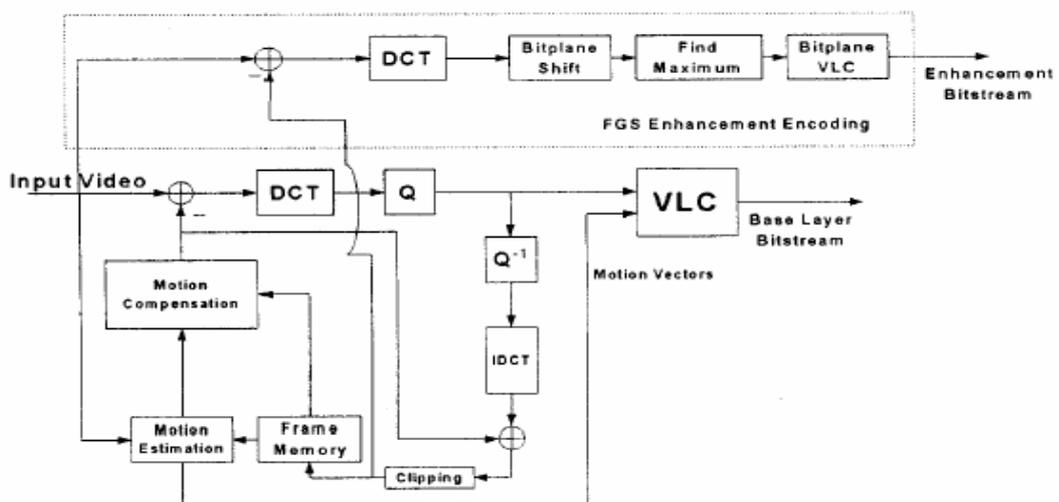


圖 2.10 FGS 編碼端架構圖[4]

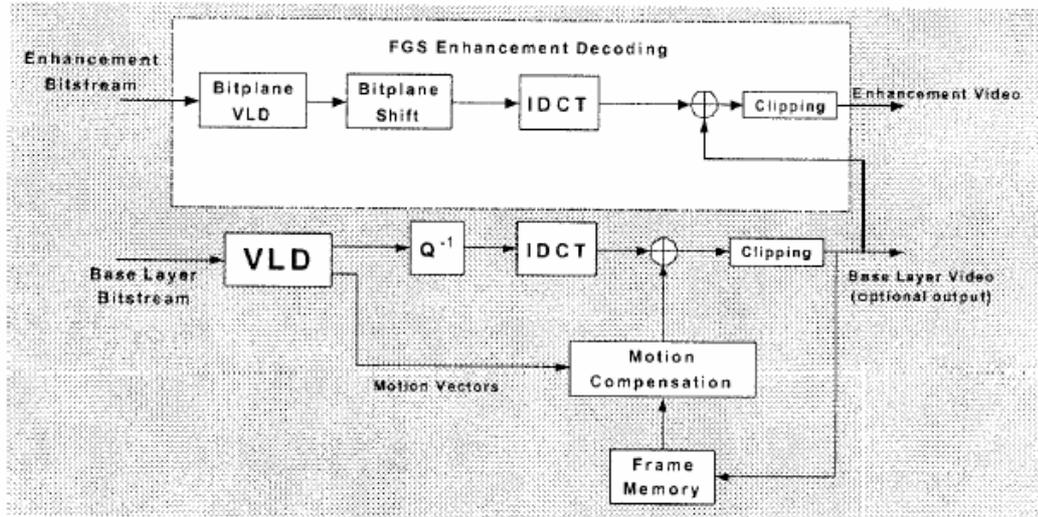


圖 2.11 FGS 解碼端架構圖[4]

而下圖則是簡明的表達出最後編碼出來的資料型式:

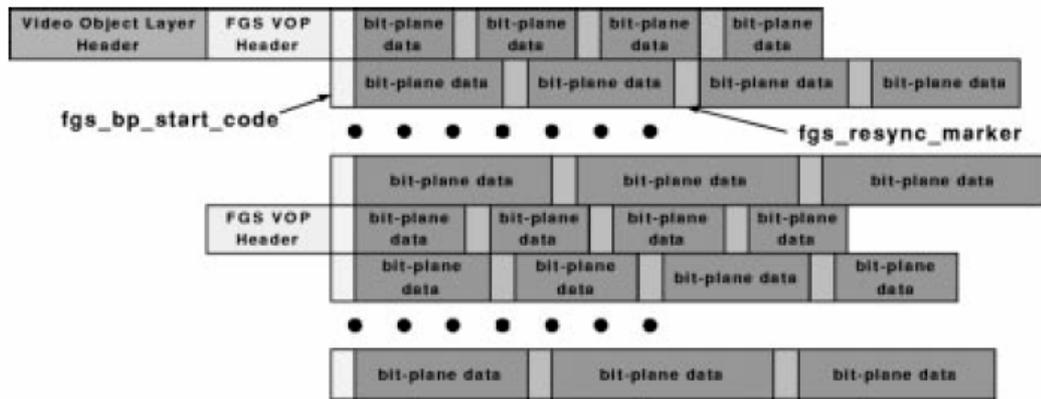


圖 2.12 FGS 資料分配示意圖[4]

下面的兩張圖則分別是 FGS 編碼和 SNR scalability 編碼以及非可調層次式編碼[non scalable coding]效能的比較，前者是一類似於 FGS 的方式，而後者則是在單一頻寬下的最佳結果。

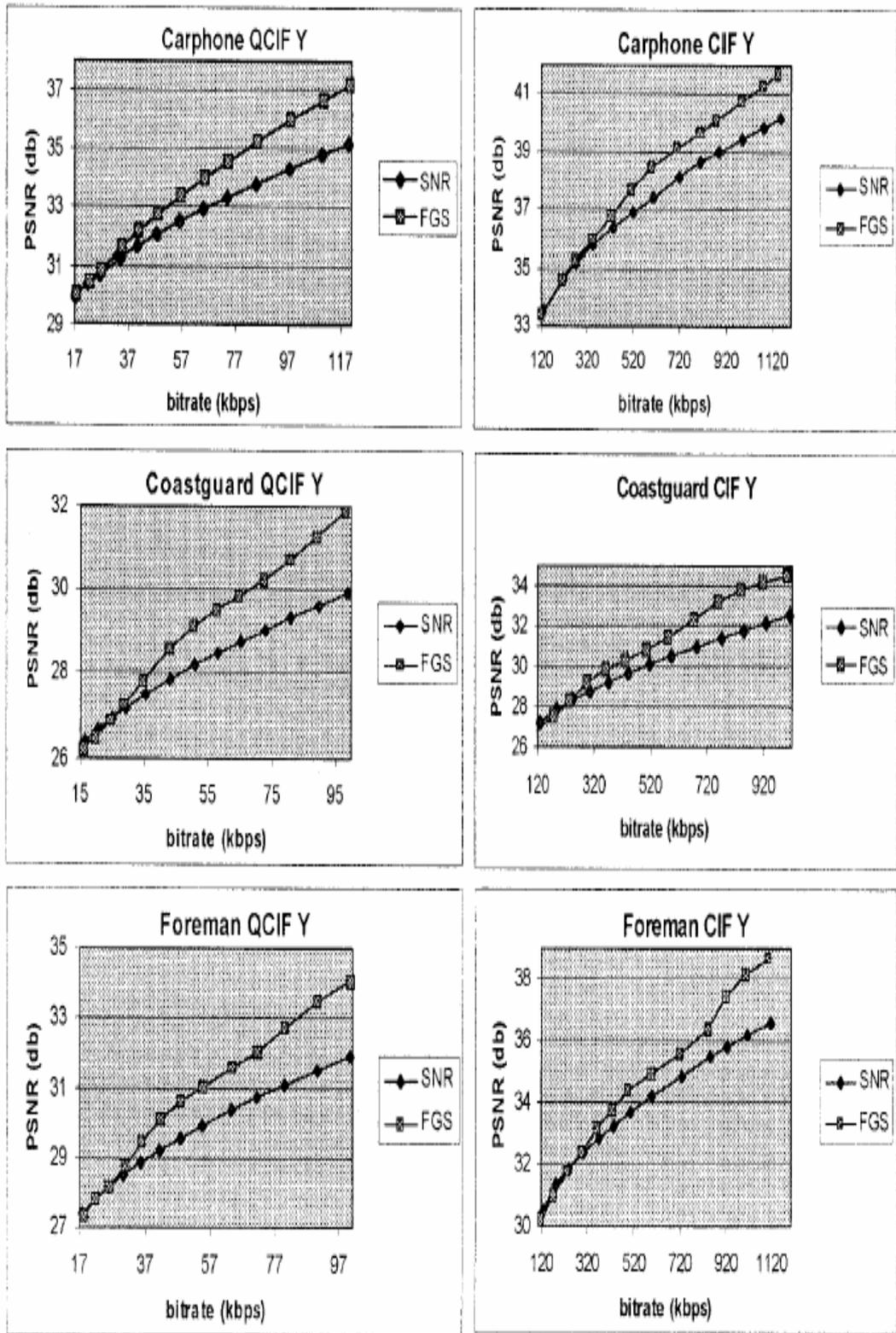


圖 2.13 FGS 和 SNR scalability 的比較[4]

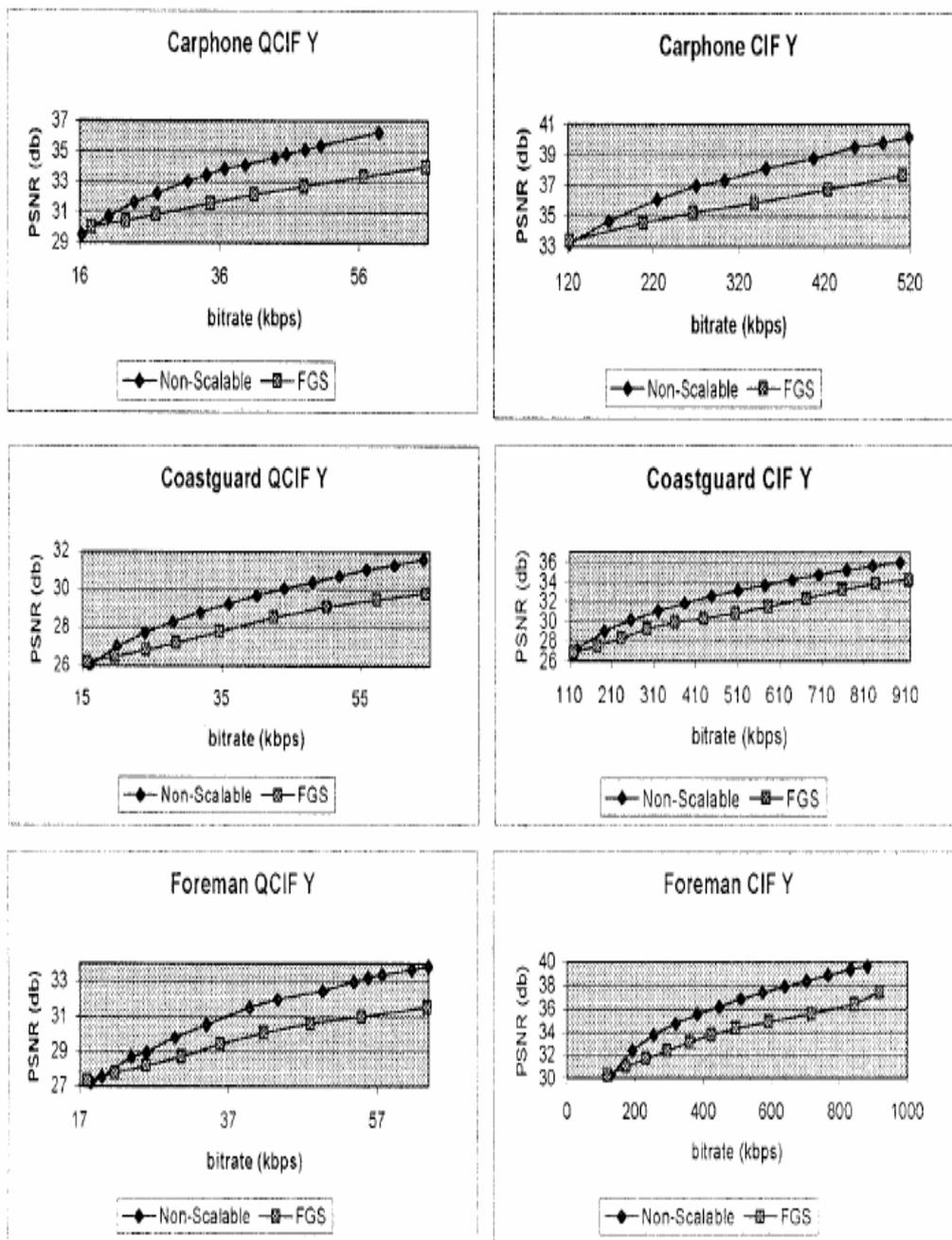


圖 2.14 FGS 和 non scalable coding 的比較[4]

FGS 編碼方式雖然提供了一個可以使得編碼可以適用於比較大頻寬範圍之解決方案，但是因為只有基本層的資料被用於運動補償[motion compensation]，而加強層的資料爲了怕沒有傳達到而帶來 error drift 的問題，所以並沒有被採用在運動補償中。明顯地，這樣的作法會降低 temporal prediction 的效率，而影響整體的編碼效能，所以一些改善的方法也相繼被提出，我們將在接下來的兩個小節中作介紹。

2.3 一些改善 FGS 的方法

在 FGS 被納入於 MPEG-4 的標準之後，很多關於改善提升 FGS 效能的方法，相繼在很多的論文中被提出，接下來將對於其中的三種方法作一個簡單的介紹，分別是 TFGS，MCFGs 和 PFGS，而其中最重要的一種改善方式 - RFGS 將在下一節中作一個比較完整的介紹。

2.3.1 TFGS (Temporal – SNR FGS)

在這個方法中主要是對於兩項 scalability 作一個結合，提供了三種可能的情形作為選擇，分別是完全採用 temporal scalability，完全採用 SNR scalability 以及同時採用兩者，可以依照當時可以被使用的頻寬、使用者的需要、解碼端的架構等，作出最適合的選擇。使用了這項技術之後，增加了編碼的可調性，由實驗結果也發現，在 PSNR 的效能上有所提升，下面是架構的示意圖：

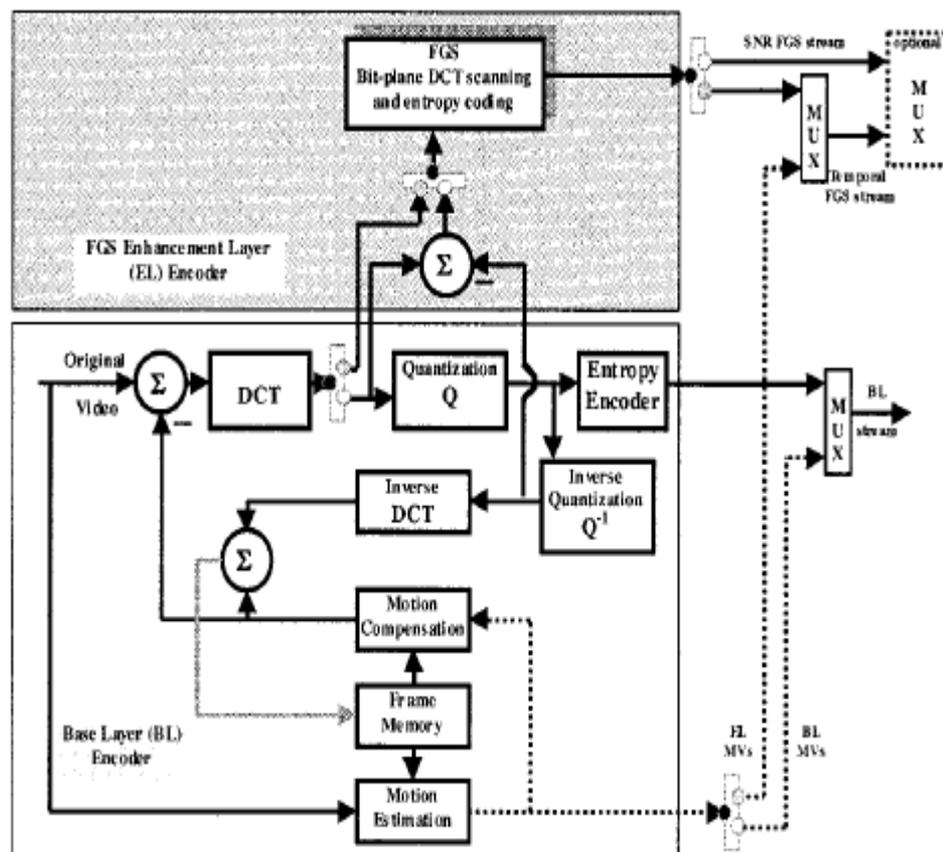


圖 2.15 TFGS 編碼端[2]

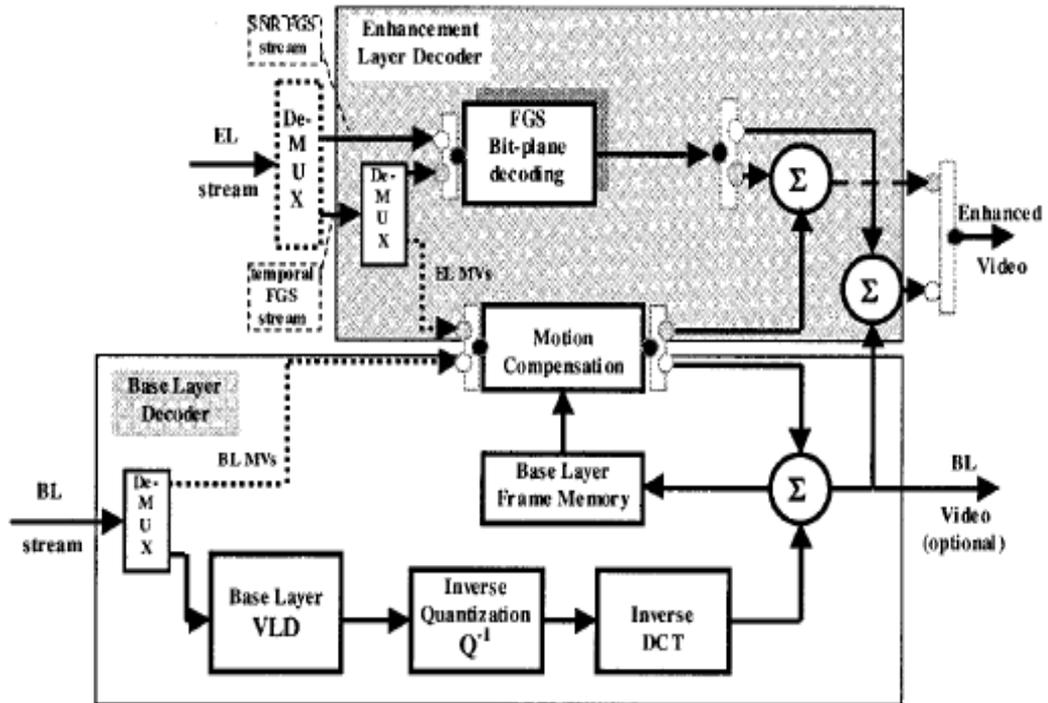


圖 2.16 TFGS 解碼端[2]

下表則是使用 TFGS 和 FGS 方法在不同的 rate 條件下的結果比較:

R_{BL} (kb/s)	R_{EL}^T (kb/s)	R_{EL}^{FGS} (kb/s)	SNR (Y/U/V) Multi-layer FGS-temporal (Figure 7)	SNR (Y/U/V) All-FGS temporal-SNR (Figure 5)
100	40	0	30.57,37.07,37.68	30.54,37.02,37.59
100	40	100	32.94,38.28,39.27	32.95,38.28,39.27
100	40	200	34.73,39.57,40.73	34.74,39.57,40.74
100	40	300	36.01,40.41,41.40	36.02,40.41,41.41
100	40	400	37.54,41.32,42.62	37.55,41.32,42.63
100	40	500	38.75,42.29,43.60	38.75,42.29,43.61
100	40	600	39.53,42.99,43.99	39.54,43.00,44.00
100	40	700	40.44,43.67,44.55	40.45,43.67,44.57
100	40	800	41.51,44.30,45.24	41.53,44.31,46.26

表 2.1 TFGS 效能比較表[2]

2.3.2 MCFGS (Motion Compensation FGS)

爲了改善原始的 FGS 架構中，不使用加強層資訊來作運動補償而帶來的編碼效率不良問題，在本方法中將提出兩種不同的編碼方式，分別是 one-loop MCFGS 和 two-loop MCFGS，並且提出了一個選擇的機制，在不同的頻寬條件下在上述兩種方法和原始的 FGS 三者中作一個選擇，改善整體的編碼效能。下圖分別爲 MCFGS 精神的示意圖，以及如何決策的方塊圖。

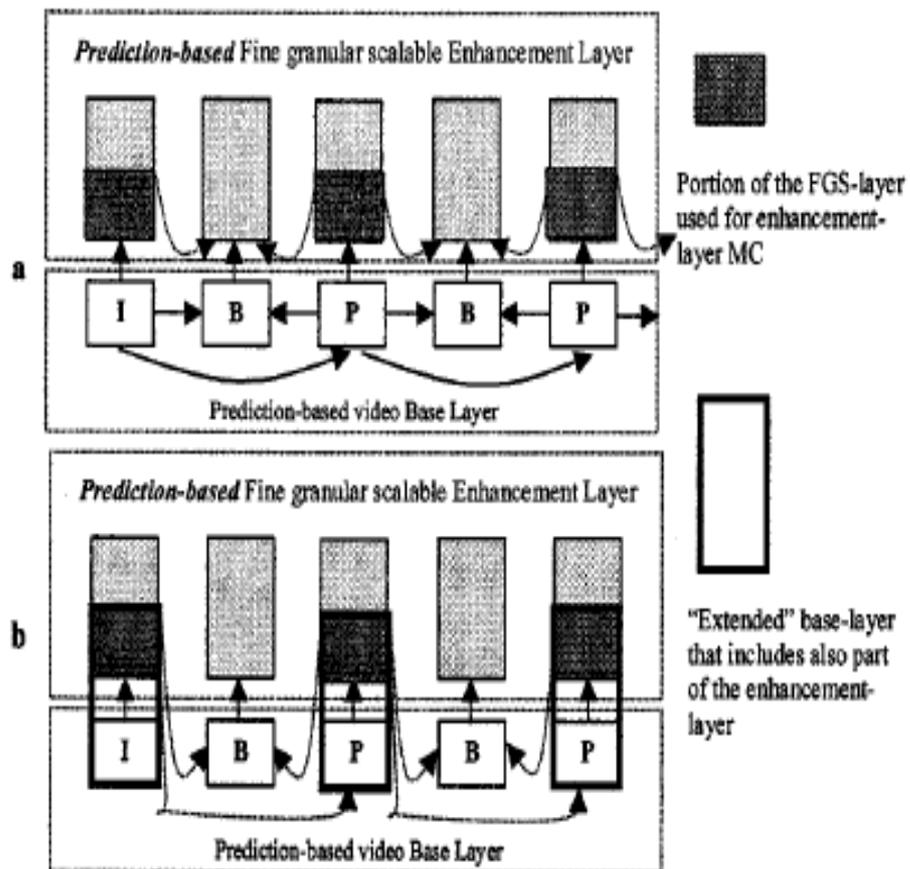


圖 2.17 MCFGS 示意圖[6]

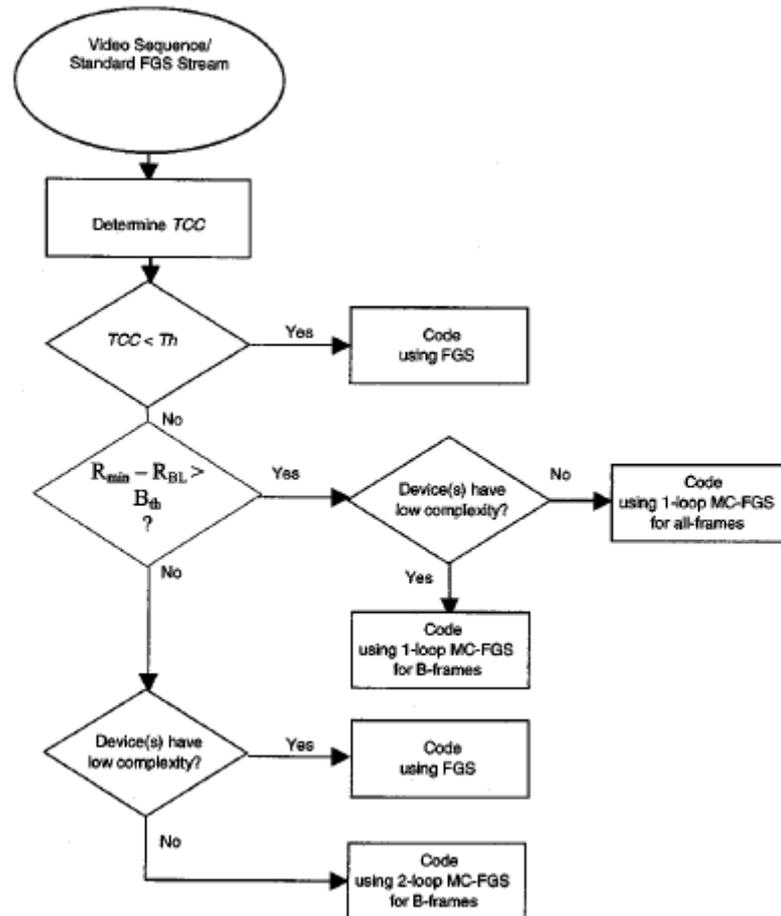


圖 2.18 MCFGS 決策方塊圖[6]

在 one-loop MCFGS 中，基本層和加強層的運動補償中，都同時採用了基本層和加強層的資訊，這樣可以大幅地提升編碼效率。但是因為基本層的運動補償也採用了可能無法獲得的加強層資訊，當頻寬不足時，將帶來嚴重的錯誤處罰，所以此方法只在 B frame 中被使用，這樣可以避免掉影響到其他張畫面的功能。

而在 two-loop MCFGS 中，則是只有在加強層的運動補償中使用加強層的資訊，這樣也可以提升部份的編碼效率，同時也可以避免掉過大的錯誤處罰，此方法可同時在 B frame 和 P frame 中來使用。

因為在加強層的編碼是使用位元編碼[bit-plane coding]，所以在在本方法中也引入了一個參數 M，來決定採用多少層的加強層資訊作為運動補償時的參考，這樣可以在編碼效率和錯誤漂移[error drift]之間得到一個平衡，採用不同的參數 M 將得到不同的編碼效果和適用範圍。

下圖是 MCFGS 和原始的 FGS 在不同的影像下，所得到的 PSNR 效果比較，圖中的 PSNR 值是採用 MCFGS 之後可以得到的改善量:

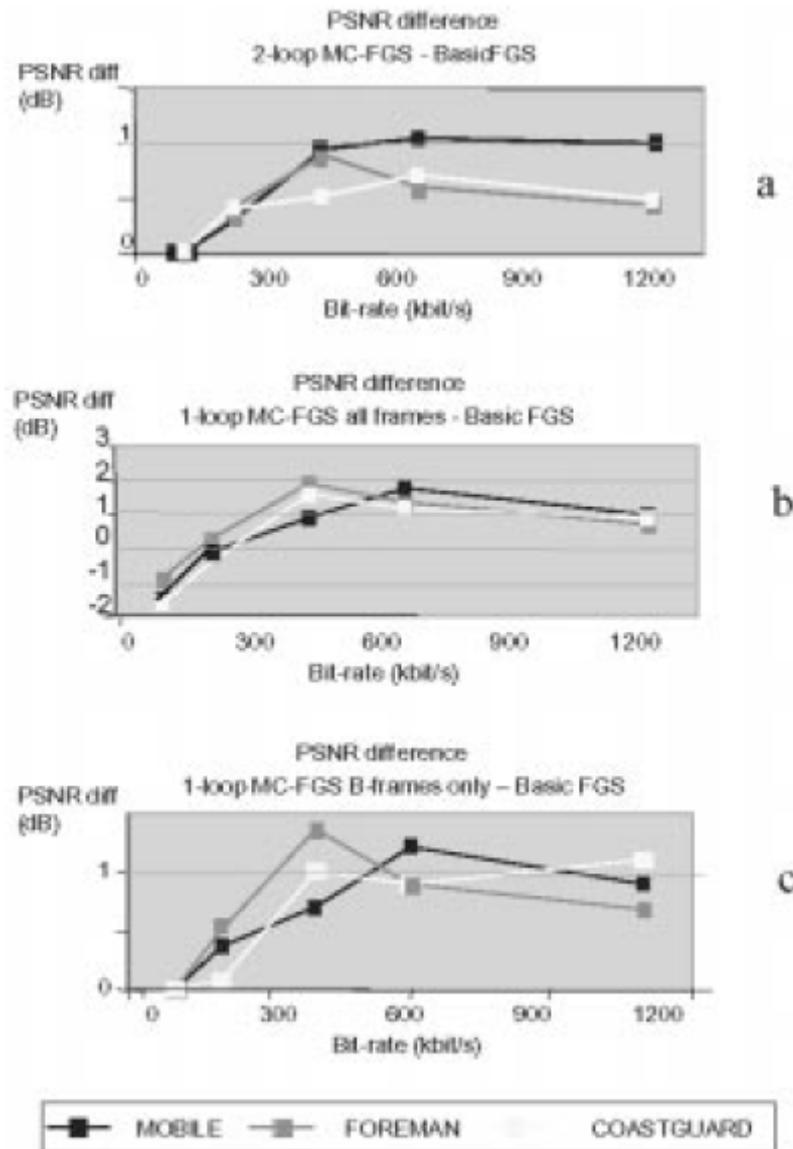


圖 2.19 MCFGs 相較於 FGS 的 PSNR 改善圖[6]

2.3.3 PFGS (Progressive FGS)

PFGS 也是對於 FGS 架構改善的一種方法，在本方法中有兩項主要的精神，分別是盡量採用最多的加強層資訊來作運動補償，和在所有的運動補償路徑中，在經過數張的畫面後，都有從基本層到最高層加強層的路徑。第一項是希望可以把編碼效率盡量的提高，而第二項則是希望錯誤漂移[error drift]的情形不會太嚴重，一次的錯誤，最多只會維持數張畫面。下圖是 PFGS 精神的示意圖：

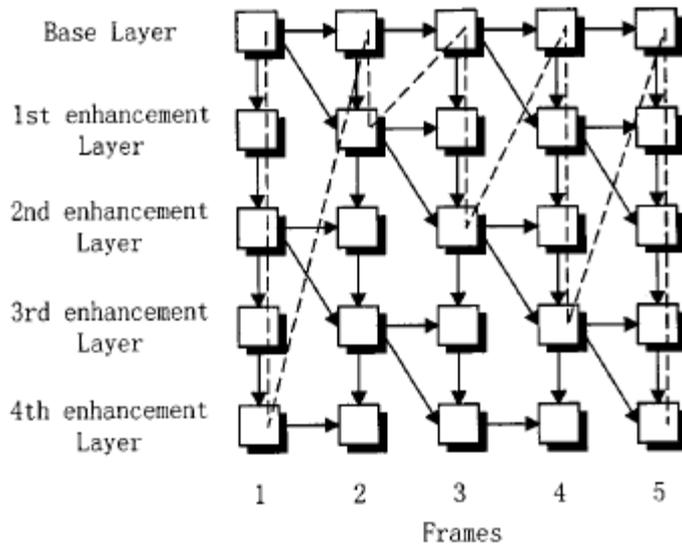


圖 2.20 PFGS 精神示意圖[3]

下圖則是 PFGS 的編碼端架構圖:

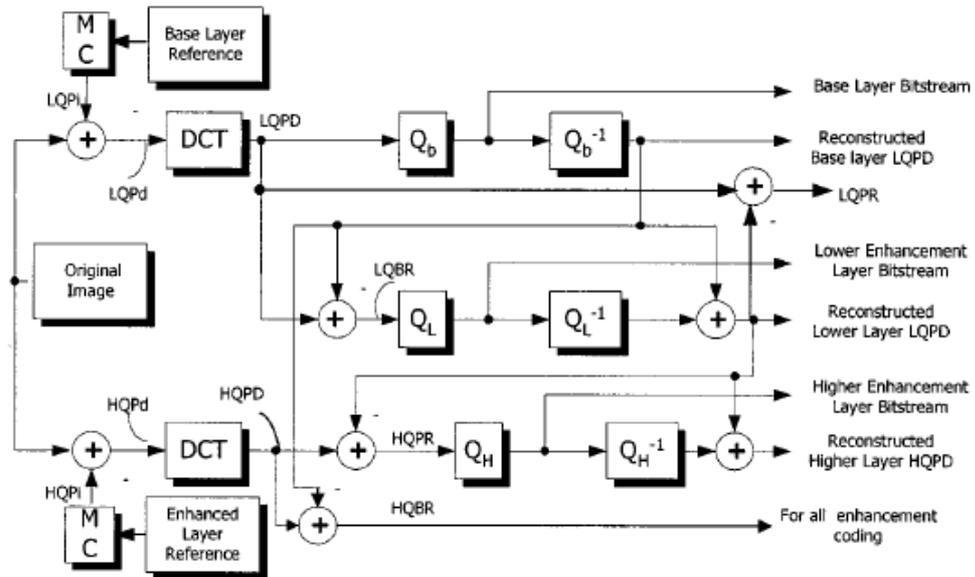


圖 2.21 PFGS 編碼端架構圖[3]

在本方法中，爲了達到第一項精神，會盡量採用更多的加強層資訊，但是如何在編碼效率和錯誤處罰中取得平衡就需要作討論。而另一方面因爲在運動補償時採用了多個可能的參考畫面，因此在硬體架構上必須使用多個不同的影像 buffer，這樣增加了一定程度的成本和複雜度。另外，對於如何維持基本層到最高的加強層之間的路徑，也有不同的設計，因此陸續有多種不同的 PFGS 版本被提出，也有各自的特色。

下面的結果示意圖對於 single layer, FGS 和 PFGS 的效能作了比較:

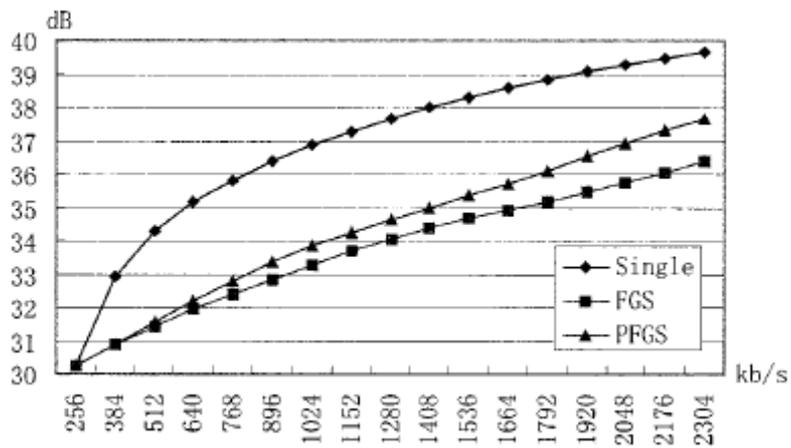
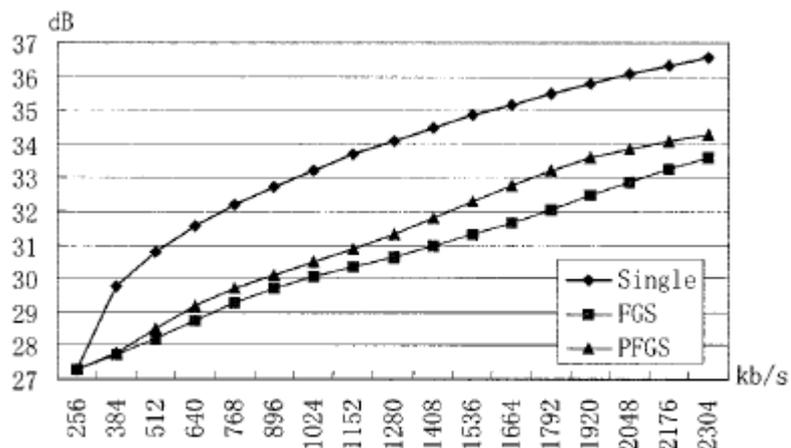
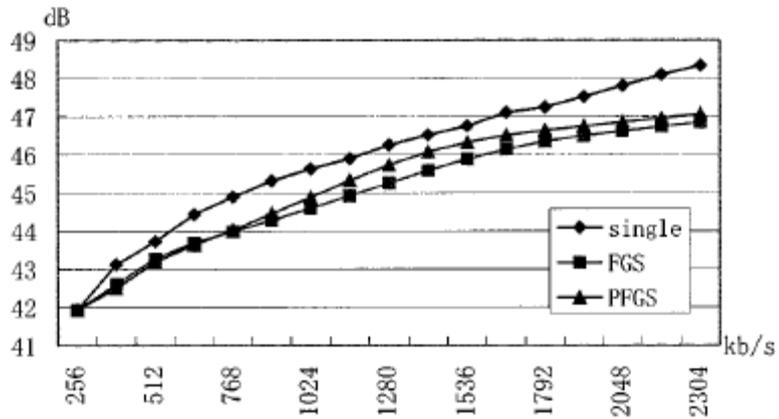


圖 2.22 PFGS Akiyo Coastguard Foreman 結果示意圖[3]

2.4 RFGS (Robust FGS)

本篇論文中所提出的方法即是以 RFGS 架構為基礎，提出一個演算法，計算出個別影像的最佳參數，因此在本節中將對於 RFGS[Robust Fine Granularity Scalable Video Coding] 作完整的介紹。

因為在 MPEG-4 標準中所訂定的 FGS，將編碼後的資料分成基本層和加強層兩個部份，但是在作 temporal prediction 時，只採用基本層的資料當作參考畫面[reference frame]，而加強層的資料部份，雖然有部份被解碼端所接受和使用，但是只能增加當張畫面的視覺效果，並不能增強運動補償[motion compensation]的效率，這樣的作法可以避免掉當頻寬不足時，所帶來的 error drift 情形，不會因為當張資料量的傳輸不足，而影響到接下來一連串畫面的影像品質。但相對地，相較於傳統的 non scalable coding，FGS 技術也犧牲了相當程度的整體編碼效率。

如同前一節(2.5)中所提及的，有不少關於提升 FGS 編碼效能的改進技術被提出，而本節中將介紹的 RFGS 是彈性最大的一種(包含兩個可以機動調整的參數)，大幅地提升了此編碼技術的適用性。

我們接下來將分別介紹兩項被本方法所採用來在編碼效率和錯誤處罰間取得平衡的技術 leaky prediction 和 partial prediction，也將介紹本方法中所提及的簡易參數猜測模式，最後並把採用 RFGS 編碼的效能和其他方法作比較。接下來的幾張圖，分別是 leaky prediction 和 partial prediction 的示意圖，以及 RFGS 在編碼端和解碼端的架構圖。

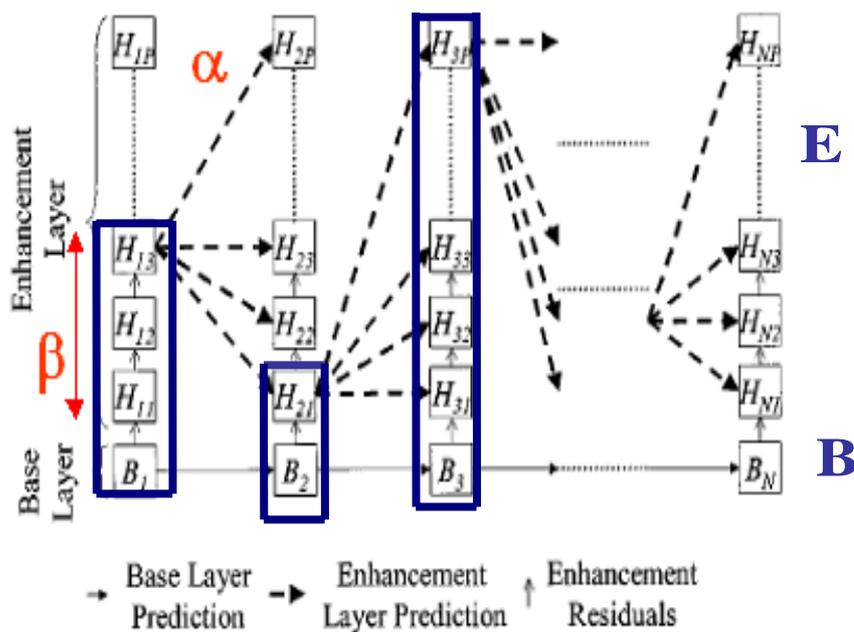


圖 2.23 RFGS 想法示意圖[7]

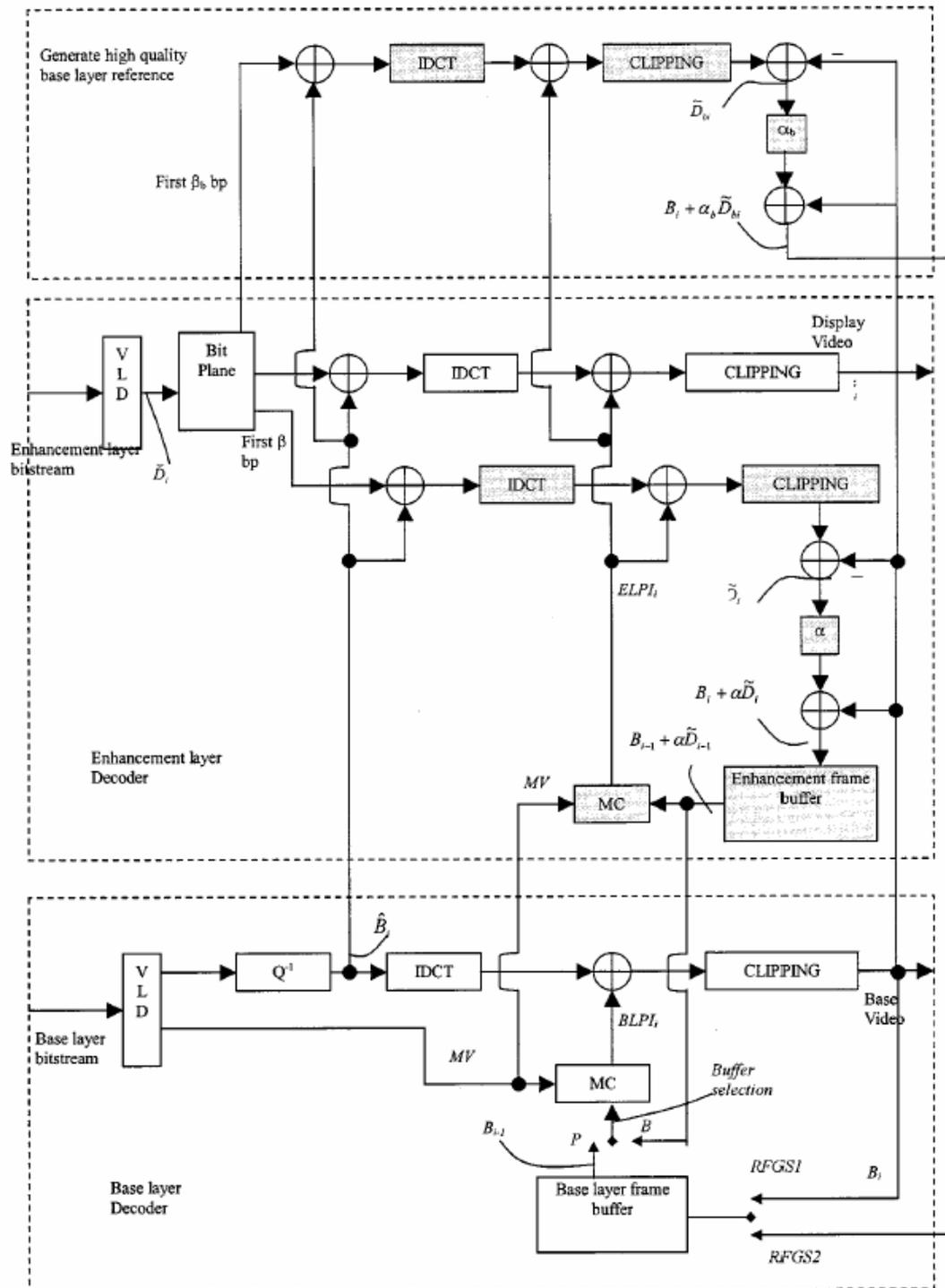


圖 2.25 RFGS 解碼端架構圖[7]

2.4.1 Leaky Prediction

在傳統的運動補償技術中，都是採用了全部的參考畫面資料量來作 temporal prediction，這樣雖然能使得編碼效率達到理論上的最佳化，但是一旦頻寬不足時，因為參考畫面中的資料量部份流失，將使得當張的影像品質，連帶地影響到接下來的數張畫面，直到下一張的 Intra Frame 出現為止。這樣子所帶來的錯誤處罰將是相當的大，因此在 RFGS 中，leaky prediction 的觀念被引用，就是只採用參考畫面的部份來作 temporal prediction。這部份引進了一個 leaky prediction 參數 α (α 的值介於 0 到 1 之間)，也就是只採用 α 倍的參考資料，剩餘的部份將編碼於 residual 之中。這樣子雖然犧牲了部份的編碼效率，由於等比級數的關係，當錯誤發生時，將使得錯誤處罰隨著畫面的進行逐漸地變小，增加了一定程度的對抗錯誤的能力。在一個比較寬的可能頻寬區間，參數 α 的引進，將提升了不少編碼的整體效能。相關的數學公式和推導，簡述如下[7]:

High quality reference image = $B + \alpha \times D$

$$F_i = (B_{i-1})_{mc} + MCFD_{BL}^i$$

$$BLPI_i = (B_{i-1})_{mc}$$

$$MCFD_{BL}^i = \hat{B}_i + Q_i$$

$$F_i = (B_{i-1} + \alpha D_{i-1})_{mc} + MCFD_{EL}^i$$

$$ELPI_i = (B_{i-1} + \alpha D_{i-1})_{mc}$$

$$\hat{D}_i = MCFD_{EL}^i - \hat{B}_i$$

$$F_i = (B_{i-1} + \alpha D_{i-1})_{mc} + \hat{B}_i + \hat{D}_i$$

$$F_i = (B_{i-1})_{mc} + \hat{B}_i + (\alpha D_{i-1})_{mc} + \hat{D}_i \\ = B_i + D_i$$

$$D_i = (\alpha D_{i-1})_{mc} + \hat{D}_i$$

$$D_i = (\alpha((\alpha D_{i-2})_{mc} + \hat{D}_{i-1}))_{mc} + \hat{D}_i \\ = (\alpha((\alpha((\alpha D_{i-3})_{mc} + \hat{D}_{i-2}))_{mc} + \hat{D}_{i-1}))_{mc} + \hat{D}_i \\ = \dots$$

$$\hat{D}_{i-2} = \check{D}_{i-2} + \Delta \hat{D}_{i-2}$$

$$\hat{D}_{i-1} = (\alpha \hat{D}_{i-2})_{mc} + \hat{D}_{i-1} \\ = (\alpha((\alpha D_{i-3})_{mc} + \hat{D}_{i-2} - \Delta \hat{D}_{i-2}))_{mc} + \hat{D}_{i-1}$$

$$\check{D}_i = (\alpha \hat{D}_{i-1})_{mc} + \hat{D}_i \\ = (\alpha((\alpha((\alpha D_{i-3})_{mc} + \hat{D}_{i-2} - \Delta \hat{D}_{i-2}))_{mc} \\ + \hat{D}_{i-1})))_{mc} + \hat{D}_i$$

TABLE I
TERMINOLOGY OF THE RFGS CODING FRAMEWORK

Notation	Definitions
F	The original image
$BLPI$	Predicted base layer frame that is generated by motion compensation from the base layer frame buffer.
$MCFD_{BL}$	Motion compensated frame difference of the base layer, which is the difference between BLPI and the original image.
\hat{B}	Coded DCT coefficients of frame $MCFD_{BL}$. The \hat{B} before de-quantization will be compressed as the base layer bitstream.
B	The base layer reconstructed image, which is the summation of $BLPI$ and \hat{B} . B will be stored in the base layer frame buffer.
$ELPI$	Predicted frame of the enhancement layer that is generated by motion compensation from the enhancement layer frame buffer.
$MCFD_{EL}$	Motion compensated frame difference of the enhancement layer which the difference between $ELPI$ and the original image.
\hat{D}	Difference signal between $MCFD_{BL}$ and \hat{B} for P -pictures or $MCFD_{BL}$ and \hat{B} for I -pictures and B -pictures. \hat{D} will be compressed as the enhancement layer bitstream.
D	The final residual used at the enhancement layer prediction loop in the encoder. $(B + \alpha D)$ will be stored at the enhancement layer frame buffer of the encoder.
\check{D}	The received \hat{D} in the decoder side. Since there may be truncation or error during the transmission of enhancement layer bitstream, \hat{D} and \check{D} may be different.
$\Delta \hat{D}$	The difference between \hat{D} and \check{D} .
\check{D}	The reconstructed D in the decoder side. $(B + \alpha \check{D})$ will be stored at the enhancement layer frame buffer of the decoder.

在 RFGS 論文中，也提出了一個線性模型[linear model]，用來挑選 Frame Based 參數 α 。其主要的精神，就是經過搜尋之後，在扣除 outlier 的影響後，發現最適合的參數 α 會和當時的 Base Layer 影像資訊之 PSNR 呈現一個線性的關係，當影像的品質愈好，PSNR 愈高時，則採用比較低的參數 α ，相反的，如果當時的 PSNR 值較低，則採用比較高的參數 α 。因為此 PSNR 值可以很容易地獲得，所以在 RFGS 論文中就是使用此一模型，來挑選合適的參數 α 。PSNR 值和最適合的參數 α 之關係圖如下所示：

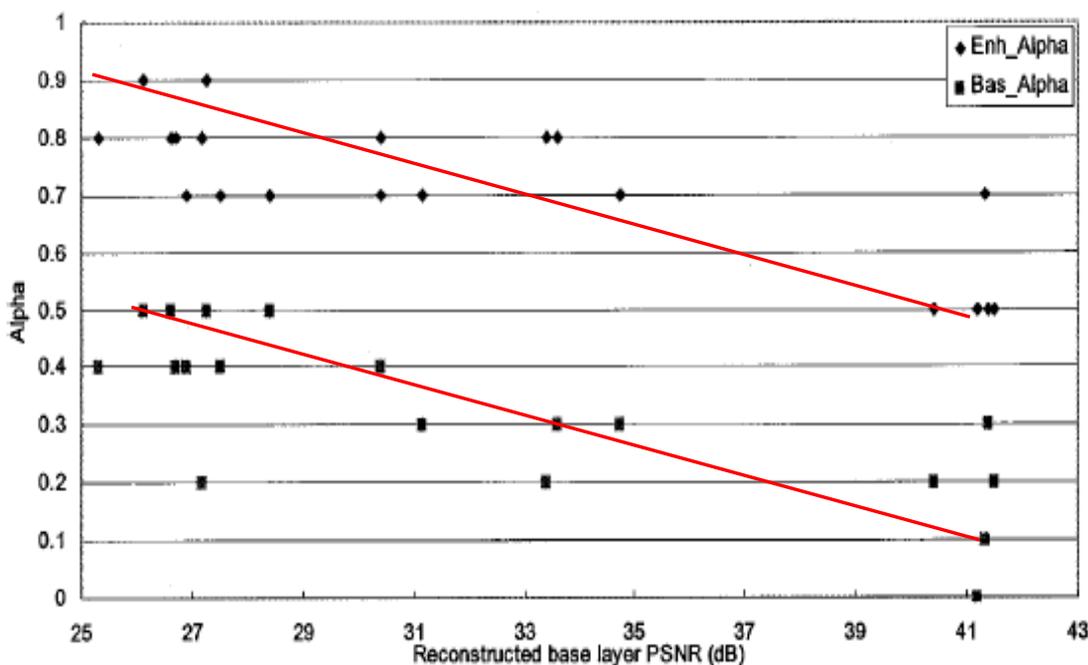


圖 2.26 RFGS 線性模型示意圖[7]

2.4.2 Partial Prediction

另外一個 RFGS 架構中的特色，就是 partial prediction 的使用。因為原本的 FGS 架構中，只採用了基本層的資訊來作運動補償，雖然可以避免錯誤的發生，但是也一定程度地影響到編碼效率，所以在 RFGS 中，將使用部份的加強層資訊，來作運動補償，而要採用多少的加強層資訊，將由參數 β 來決定(參數 β 的值介於 0 到最大層之間)。如果採用的 β 層加強層資訊在解碼端順利地取得，可以增加一定程度的編碼效率；但是相反地，如果這 β 層的加強層資訊，無法在解碼端被取得，則不但沒有達到增加編碼效率的效果，還會產生錯誤，如果沒有適當的機制來加以彌補，還會影響接下來一連串影像畫面的效果(error drift)。所以如何採用適合於當時頻寬狀況的 β ，對於最後的整體編碼效能將有很大的影響，在該篇論文中，也依據實驗的結果，對於不同可能頻寬大小，提出了挑選參數 β 的方法：當頻寬狀況比 512Kbits 還要差時，將挑選 $\beta = 2$ ；當可能的頻寬狀況在 256Kbits 到 1.2Mbits 之間時，將採用 $\beta = 3$ ；最後，當頻寬狀況更好時，將採用 $\beta = 4$ 。

2.4.3 實驗結果

接下來的幾張圖，將分別表示出 RFGS 和 FGS 以及 Single Layer 在不同的頻寬狀況下的編碼效果，所採用的測試影像分別是 Akiyo，Foreman 和 Coastguard。

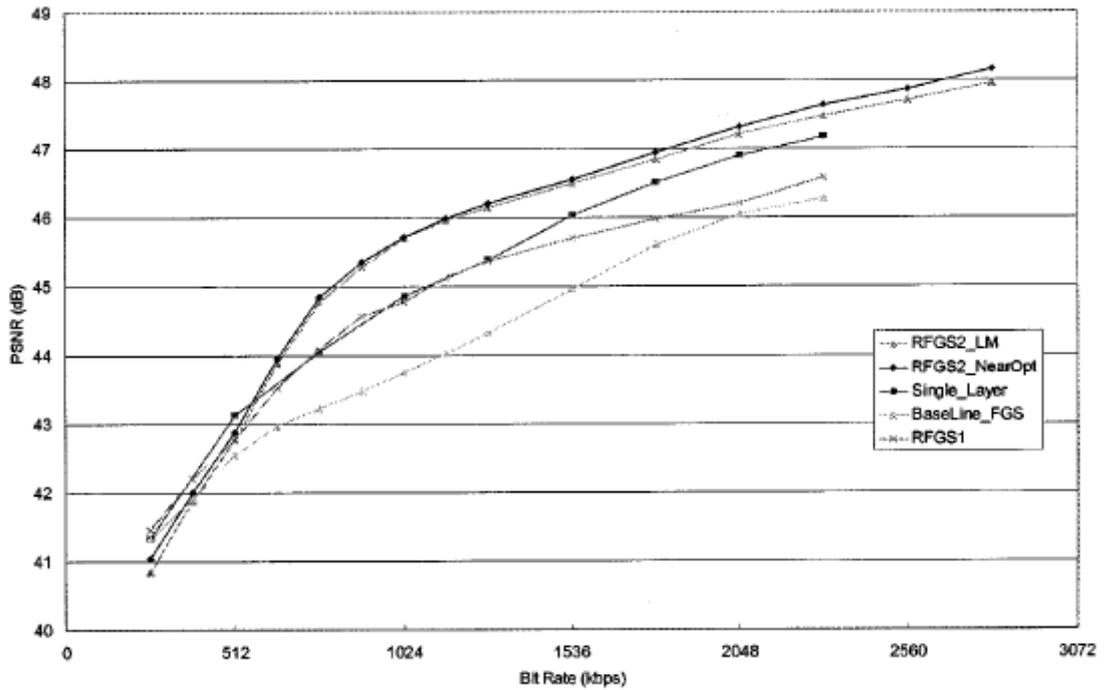


圖 2.27 Akiyo 實驗結果[7]

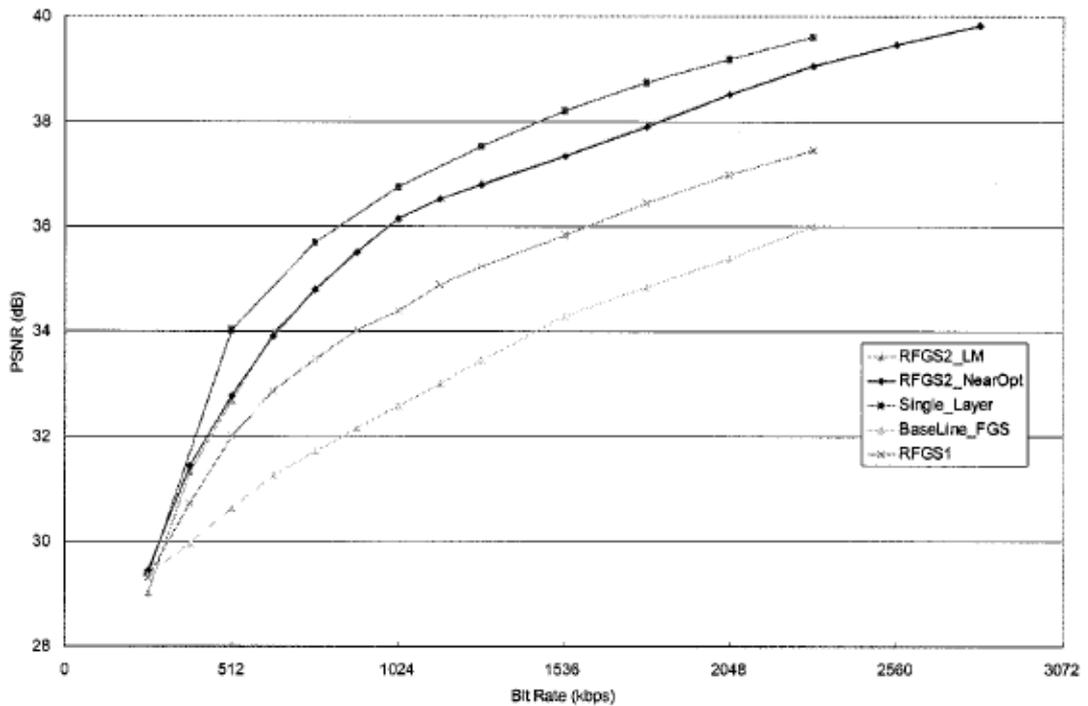


圖 2.28 Foreman 實驗結果[7]

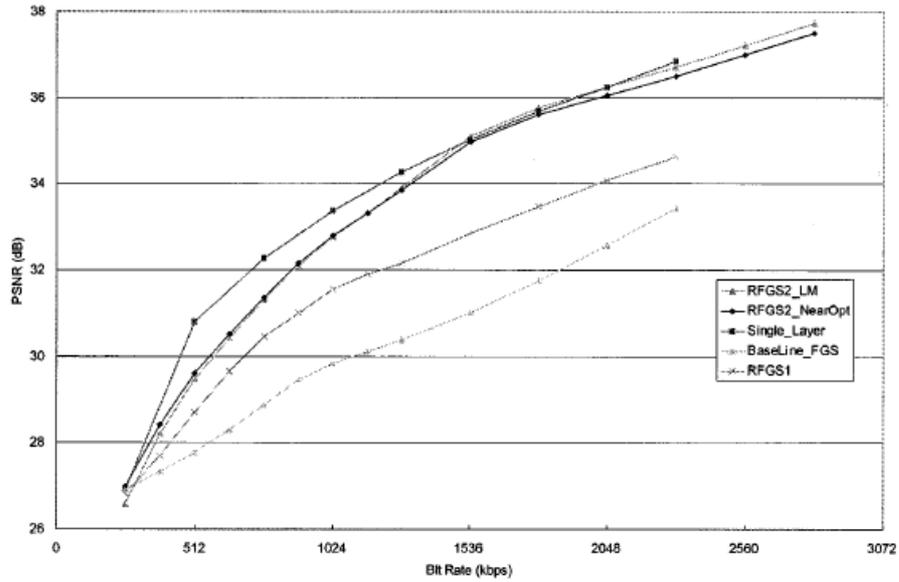


圖 2.29 Coastguard 實驗結果[7]

接下來的兩張結果示意圖，則是 RFGS 和 PFGS 在不同的 MPEG 測試環境下，採用 Foreman 和 Coastguard 所獲得的影像品質比較圖。

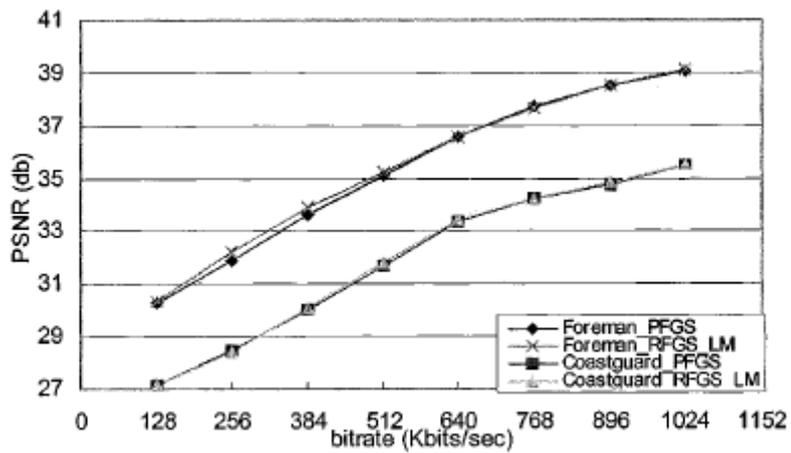
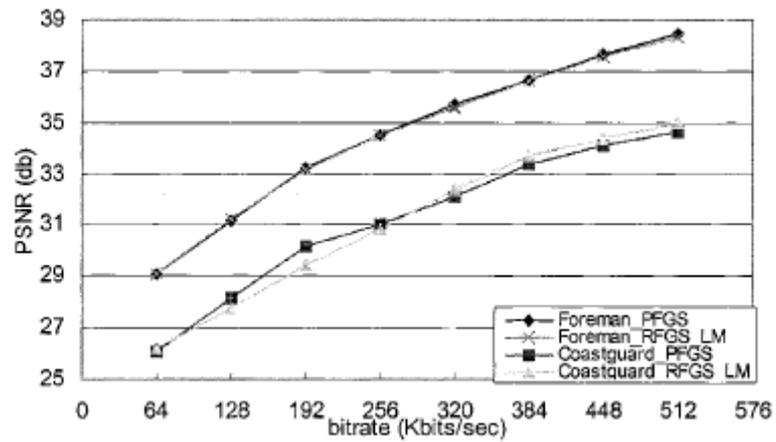


圖 2.30 RFGS 和 PFGS 的結果比較[7]

第三章 RFGS 最佳化參數挑選

在本章中，第一節我們將陳述之前提出之 RFGS 參數最佳化的方法，第二節將介紹以 MSE 替代 PSNR 為比較基礎的方法，第三節將比較兩種方法所得到的結果和適用性，最後在第四節中，我們將對於本方法作一個總結。

3.1 先前的 RFGS 最佳化參數挑選方法

因為 RFGS 的架構中，提供了兩個可變的參數，所以本方法中將取 RFGS 的架構為基礎，發展出一套尋找每一個 FRAME 每一個 MB 的最佳參數。在本節中，我們將完整地介紹之前 RFGS 參數最佳化的過程，依序是問題的定義、FRAME_BASED 最佳參數的找尋、MB_BASED 最佳參數的找尋、以及三階段搜尋法[Three-Step Search]。

3.1.1 問題的定義

我們將在一段可能的頻寬中，找尋可以帶來最好效果的一組參數，我們將把這一段的頻寬分成數個區域，並依此產生N個檢查點[check point]， S_1, \dots, S_N ，其中如果 $i < j$ 則 $S_i < S_j$ 。如同前面章節中對於RFGS的介紹，temporal prediction的效能在不同區域的頻寬將會有不同的表現，在比較高的 S_i 時，如採用較多的加強層[EL]資訊，將可以使得temporal prediction的效率提高，使得motion compensation時的gain G_i 變大，進而使得整體的效能提升;相對地，當在比較低的 S_i 時，如採用過多的加強層[EL]資訊來作temporal prediction，不但無法提升效率，還會產生error drift的情形，帶來比較大的lost L_i ，而使整體的效能下降。可調層次式編碼[scalable coding]就是為了在一段頻寬中帶來最大的編碼效率[coding efficiency]，在我們提出的方法中，將以N個檢查點[check point]的編碼效率的平均，作為編碼效能(coding performance)。

如前所述，在 RFGS 的架構中，提供了兩個可供調整的參數 α 和 β 來控制 temporal prediction，其中 β ($0 \leq \beta \leq \max_bitplane$)是指有多少層的加強層資訊將被用來作 motion compensation，而 α ($0 \leq \alpha \leq 1$)則是指有多少比例的資訊將被用來作 motion compensation，可以達到控制 error drift 的效果。

先前所提出的方法的目標就是找到一組的 α_{opt} 和 β_{opt} ，能夠在給定的頻寬區域中得到最好的效能，可以用下面的式子來表示：

$$(\alpha_{opt}, \beta_{opt}) = \arg \max_{\alpha, \beta} \left(\frac{1}{N} \times \sum_{i \in S} (G_i + L_i) \right) \quad \dots 3.1$$

在這個方法中，將以 PSNR 作為一個客觀的影像品質評量標準，而 PSNR 的定義如下：

$$PSNR = 20 \times \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \quad \dots\dots 3.2$$

其中 MSE 指的是原始影像跟經過編解碼所得到的影像的平均差平方(Mean Squared Error)。

3.1.2 FRAME_BASED 最佳參數

在最基本的 RFGS 架構中，是依照當時最可能的頻寬來決定參數 β 。至於參數 α 的部份則是根據當時的 PSNR 值，依照一個 linear model 來決定。在參數決定的過程中，並沒有採用個別影像特質的資訊。而在先前所提出調整 α 與 β 參數的方法中，雖然最終的目標是決定 MB_BASED 的參數，但是為了避免找到的最佳參數值只是局部最佳[local maximum]解，所以在開始找尋 MB_BASED 的參數最佳值之前，會先找出 FRAME_BASED 最佳參數，之後再依此為初始值，接著找出 MB_BASED 的最佳參數。如此不但可以避免掉局部最佳的情形，也可以有效節省 MB_BASED 搜尋的時間。

在該方法中，選擇 β 為 2、3、4、5、6 等五種情形， α 為 0.96875、0.90125、0.8125、0.71875、0.59375、0.5、0.40625、0.28125、0.1875 等九種情形，在一共四十五組可能的情況下，分別測試在 N 個檢查點之下影像的 PSNR 值，取出平均值後，再挑選最好的一組 α 、 β 值，作為此張影像的 FRAME_BASED 最佳參數。如果增加 α 的測試值數量，可以使得所找到的最佳參數更為精確。

3.1.3 MB_BASED 最佳參數

以 CIF 格式的影像而言 在同一張影像中包含了 396 個 MB，在不同的區域中，可能有著完全不同的影像特質，如影像運動速度、影像複雜度、位置等等，使得不同 MB 中所需要被編碼的 residual 會有一定程度的差異。有鑑於此，在本方法中決定給予每個 MB 不同的參數值，藉著對於 MB 之間參數值的微調，提升整體的編碼效能。接下來，將簡要介紹 MB_BASED 最佳參數微調的過程。

在決定了 FRAME_BASED 的參數之後，就以此為 MB_BASED 參數的初始值來進行微調。首先定義 N 個檢查點在解碼端的 PSNR 值之平均為 APR(average PSNR)，在接下來的最佳化過程中，都將以 APR 作為效能好壞的依據。以下將以 β 為例子，解釋如何找到最佳參數的過程。首先，定義 TR(j)(gain trend)表示 β 值的改變趨勢：

$$\begin{aligned} TR(j) = & 0 & \text{if } \max(APR(\beta_0(j)^+), APR(\beta_0(j)^-)) < APR(\beta_0(j)) \\ & +1 & \text{else if } APR(\beta_0(j)^+) > APR(\beta_0(j)^-) \\ & -1 & \text{otherwise,} \end{aligned} \quad \dots\dots 3.3$$

其中 $\beta_0(j)$ 表示此 MB 的初始 β 值， $\beta_0(j)^+$ 和 $\beta_0(j)^-$ 分別表示將 $\beta_0(j)$ 往上和往下調整一個 step 的 β 值。下圖的例子可進一步的作為說明：

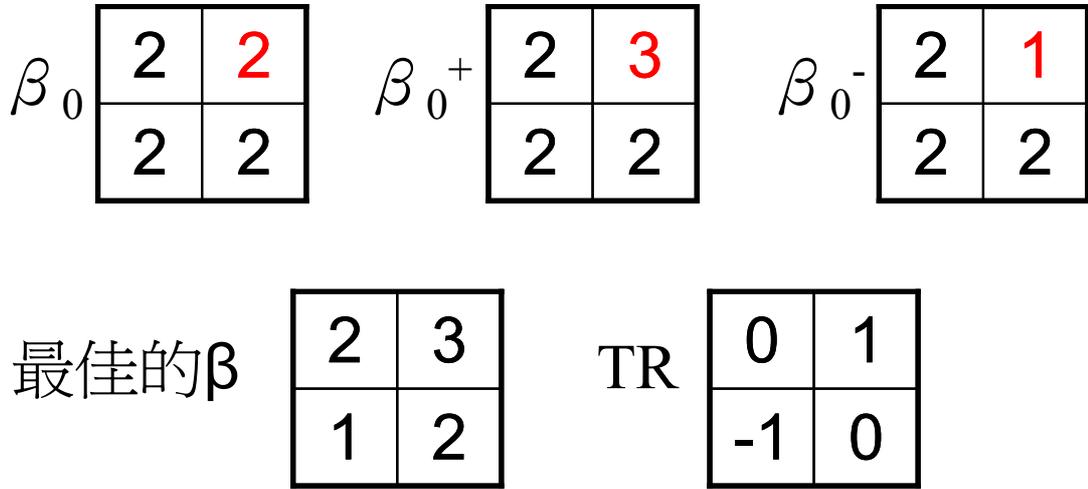


圖 3.1 TR 示意圖

因為之前的計算都是在每個 MB 彼此獨立不會互相影響的前提下，但是在實際的狀況中，並非如此，MB 參數值的調整也會或多或少地影響到其他 MB 參數值的決定，特別是相臨近的 MB。也就是說，並不是依照 TR 把每一個 MB 都作調整就可以帶來最佳的效能，為了方便之後的運算，讓對於 PSNR 有比較大貢獻的 MB 有比較高的優先順序，將作以下的運算。接下來，定義 $LG(j)$ 為只對 j th MB 作相對應於 $TR(j)$ 調整後，可以對 APR 帶來的增益，在算完所有的 $LG(j)$ 後，再將 LG 按照值的大小，由大到小重新排列，產生一組新的 $LG'(j)$ ，同時也產生相對應的 $TR'(j)$ 。另外定義 OD 表示優先順序，下圖的例子可作為說明：

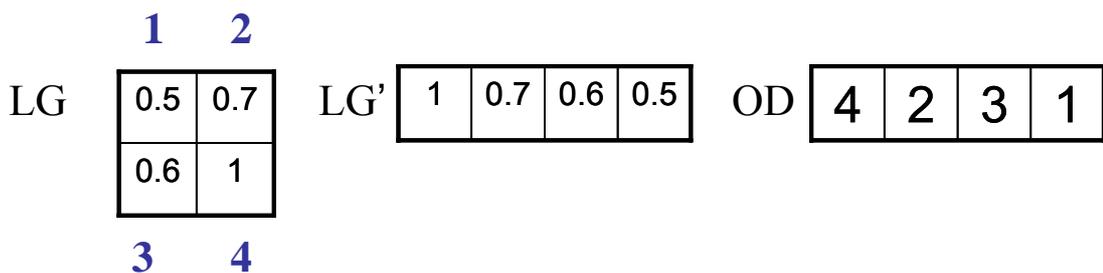


圖 3.2 LG OD 示意圖

最後，依照先前的想法，將利用 TR 和 OD 來產生一系列的測試組合，如果在一張 FRAME 中有 K 個 MB，那麼將產生一共 K 組的測試組合，在第 j 組測試組合中，對於貢獻較大的 j 個 MB 作調整，其餘的 MB 參數則不予變動，最後分別對這 K 組的組合進行測量，取出能夠帶來最大 APR 的測試組合作為最佳參數，下圖的例子可作為說明：

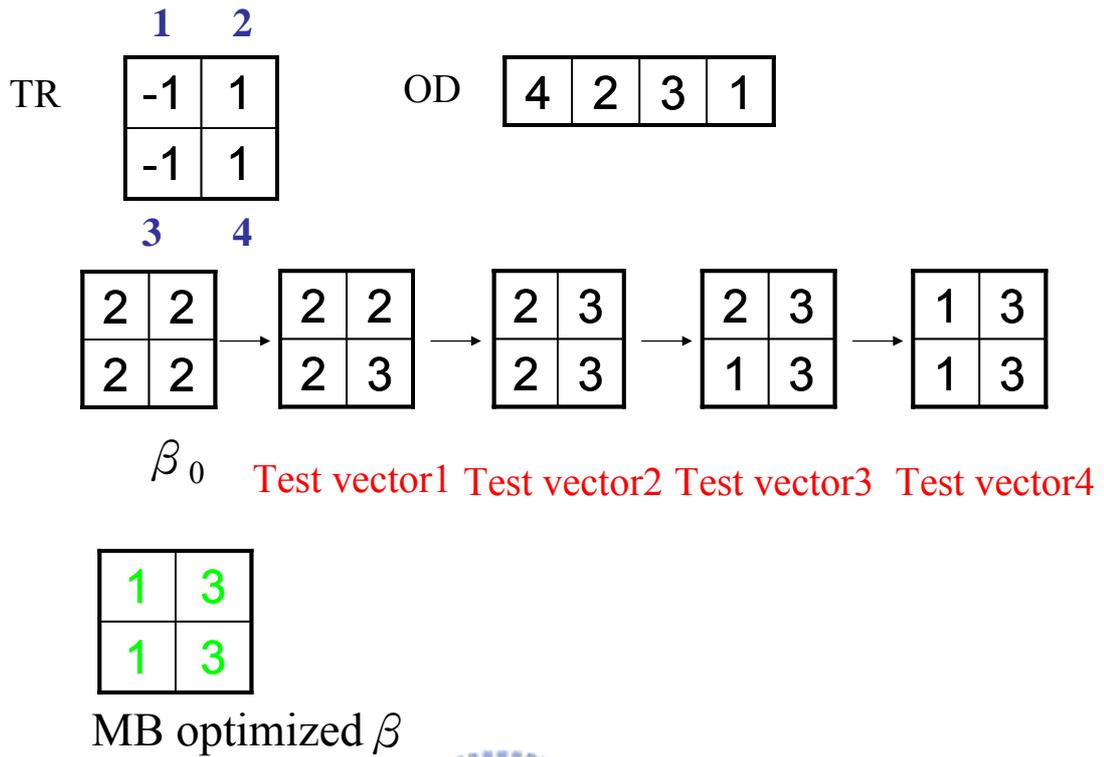


圖 3.3 Test Vectors 示意圖

以 K=396，也就是每張 FRAME 有 396 個 MB 的 cif 影像為例，表示出其 396 組測試組合所帶來的 APR，由下圖中可以明顯地發現，並不是調整所有的 MB 才能帶來最佳的效能，而產生最佳效能的測試組合(峰值的位置)，會隨著影像的不同而改變。

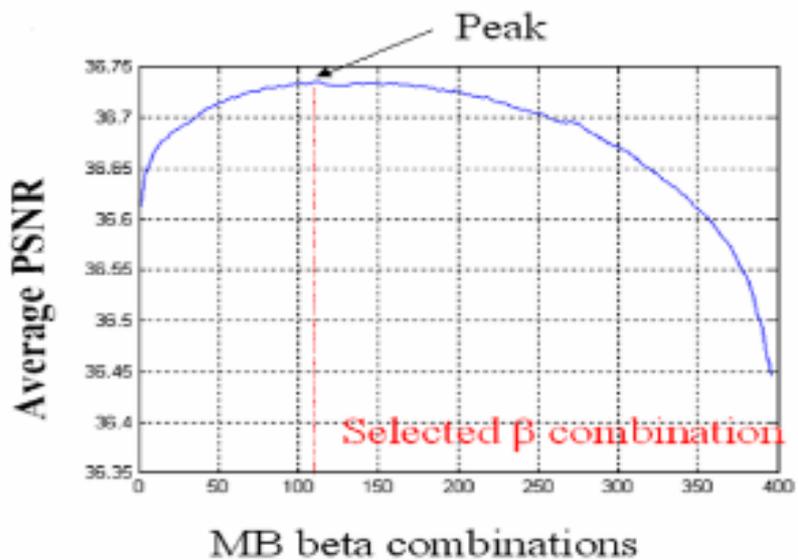


圖 3.4 Test Vectors 結果示意圖[10]

3.1.4 三階段搜尋法 Three-Step-Search

在前面幾小節中分別介紹了FRAME_BASED和MB_BASED的最佳參數找尋方式，而在本節中的三階段搜尋法則是把兩者作一個結合，同時考量 α 、 β 所帶來的影響，搜尋出最佳的參數組合。因為在實際的測試結果中，發現當進行第三次之後的MB_BASE調整，對參數組合 α 、 β 的改變已經相當有限，而且幾乎沒有帶來更多的PSNR增益。相對之下，第二次的MB_BASED調整可以對於第一次的MB_BASED調整結果，提供一定程度的PSNR改善，所以最後本方法是採用一次的FRAME_BASED和兩次的MB_BASED調整，來得到最後的最佳參數 α_{opt} 、 β_{opt} 。詳細的步驟如下：

1. 爲了增加搜尋組合的廣度，也爲了避免局部最佳的情形發生，先進行FRAME_BASED最佳參數的找尋，得到能夠獲得最佳PSNR值的參數組合 α_0 、 β_0 。

2. 接下來，把步驟1中所得到的 α_0 、 β_0 作爲起始點，把 α 、 β 分別調上調下一個step加上不變動，得到一群組合(a, b)， $a \in (\alpha_0(j)-SP, \alpha_0(j), \alpha_0(j)+SP)$ ， $b \in (\beta_0(j)-SP, \beta_0(j), \beta_0(j)+SP)$ 。在一共九種可能的組合中找出能得到最大的APR一組，並且依照之前的方法算出TR和LG，接著算出TR'，LG'和OD，所得到的參數組合爲 α_0' 、 β_0' 。

3. 重複步驟2中的程序，把初始值換成 α_0' 、 β_0' ，經過運算就可以得到新的一組參數 α_0'' 、 β_0'' ，也就是最後的最佳參數 α_{opt} 、 β_{opt} 。

3.2 MSE 最佳參數

在本節中將提出一個改以MSE大小爲基礎的最佳化參數挑選方式，取代原本的以PSNR大小爲基礎的方法。因爲MSE和PSNR之間的關係並非線性，在比較低的PSNR時，隨著MSE的下降，PSNR上升的速度較慢，而在比較高的PSNR時，隨著MSE的下降，PSNR將帶來可觀的上升。但是視覺的品質並不同於PSNR，反而是採用MSE的大小較有代表性，因此在考量影像品質時，在一段可能的頻寬中有N個檢查點，如果如同之前的方法，取各個不同頻寬下的PSNR作平均的話，將使得高頻寬的檢查點過度地影響參數的選擇，雖然挑選出來的參數組合在比較高的頻寬下，有較好的表現，但是將對於低頻寬點帶來嚴重的影響。有鑑於此，我們提出了改以MSE爲衡量標準的方式，使得各個頻寬點的重要性較爲平均，提升整體平均的影像品質。下面的示意圖表示出了PSNR大小和MSE大小之間的關係：

$$PSNR = 20 \times \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \quad \dots 3.4$$

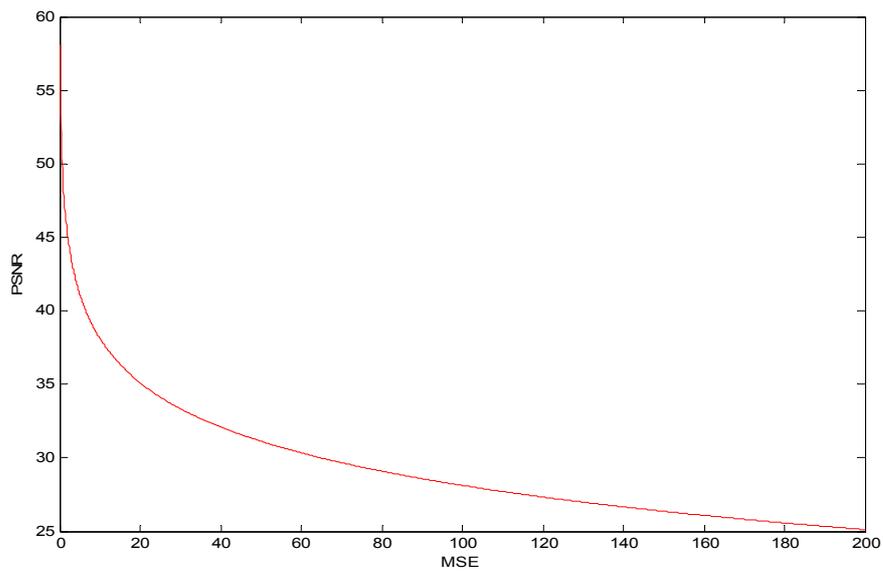


圖 3.5 PSNR 和 MSE 關係示意圖

在相同差距的 PSNR 時，將因為高低 PSNR 而帶來不同的視覺品質差量，其結果如下圖所示:



(A)



(B)



(C)



(D)

圖 3.6 不同 PSNR 值時的 PSNR 影響

以 MSE 為大小為基礎的方法和以 PSNR 大小為基礎的方法大致相同，只是從原本的挑選較大的 PSNR 平均，改成挑選較小的 MSE 平均而已，其他的步驟和作法都同於之前的方法，在此也不再重複。

明顯地，當只考量單一個頻寬時，兩個方法挑選出來的最佳參數組合將是相同的，但是當考量的是一整段的頻寬時，由上述的 PSNR 和 MSE 關係示意圖，可以發現，兩個方法所挑選出來的最佳參數組合將變得不一樣。我們認為採用較貼近於視覺品質的 MSE 作為比較標準，將可以帶來比較好的結果，所以在本論文所提出的方法中，也將以 MSE 的大小作為挑選最佳參數的依據。

3.3 兩個方法的結果比較

在本節中，將以 FRAME BASED 的結果為例子，比較在同樣的頻寬區域下，採用相同的檢查點，使用兩種不同的方法所找出的結果有什麼不同。我們將以 news、foreman、stefan 這三個測試影像，分別代表慢速、中速和快速的影像，測試的頻寬從 0Kbyte 到 2560Kbyte，中間取了九個檢查點，分別是 0Kbyte、256Kbyte、384Kbyte、512Kbyte、768Kbyte、1024Kbyte、1536Kbyte、2048Kbyte、2560Kbyte 的加強層資料，外加 256Kbyte 的基本層資料。這樣的測試情況包含了好的頻寬狀況、中等的頻寬狀況和不佳的頻寬狀況。

我們產生了一組考量整個區段的頻寬之參數，使用這組參數，我們測試出在不同的頻寬條件下，經過編碼、傳輸、解碼影像的品質，用 PSNR 和 MSE 兩種不同的標準來檢視。雖然用 PSNR 和 MSE 兩種不同的標準來選擇參數，但是目前多數的論文是採用 PSNR 為衡量標準，因此在本論文中的結果比較圖中，仍將以 PSNR 值為效果好壞的比較標準，所得到的結果如下圖所示：

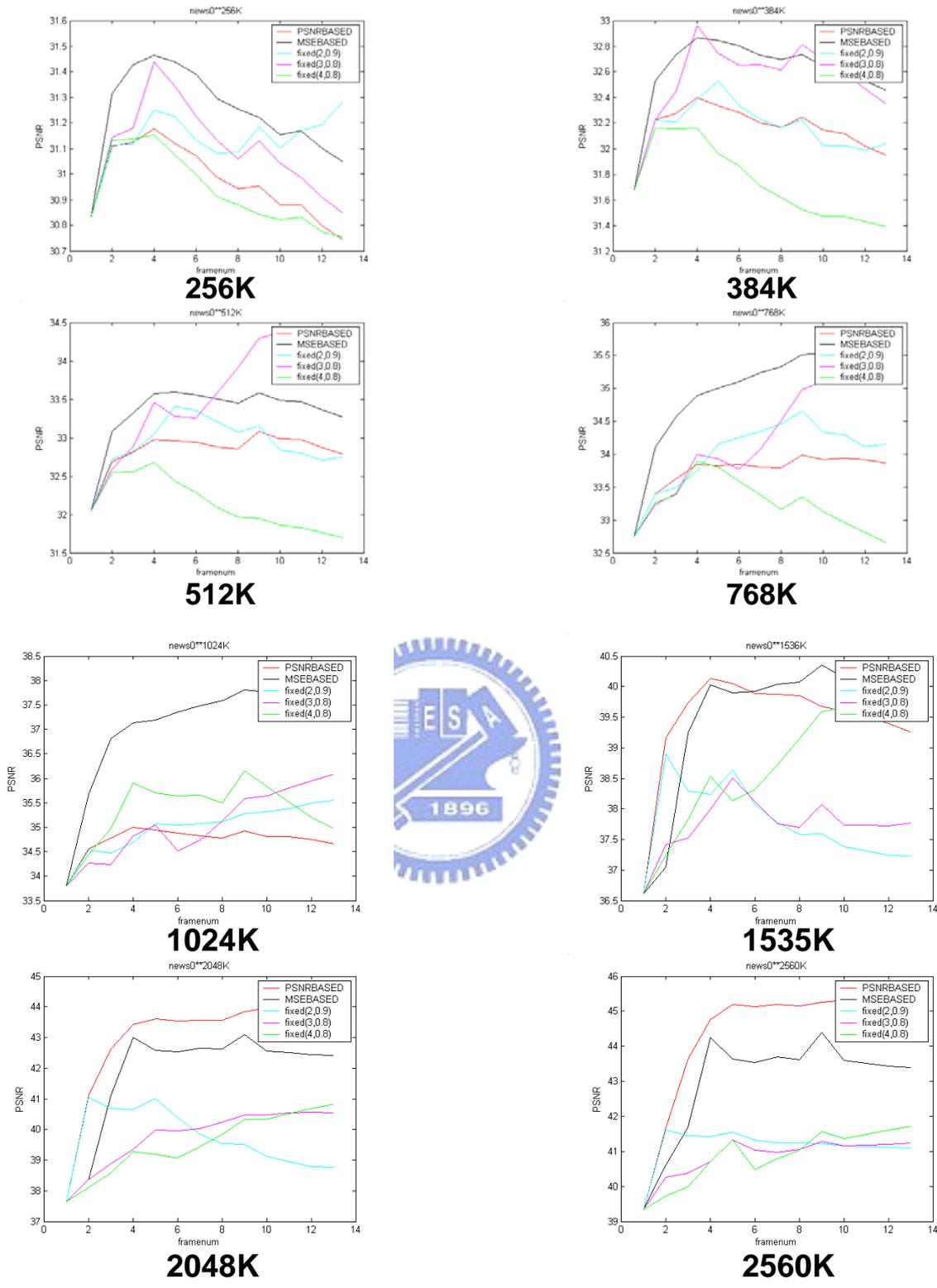


圖 3.7 news frame-based 結果

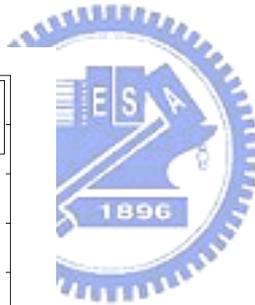
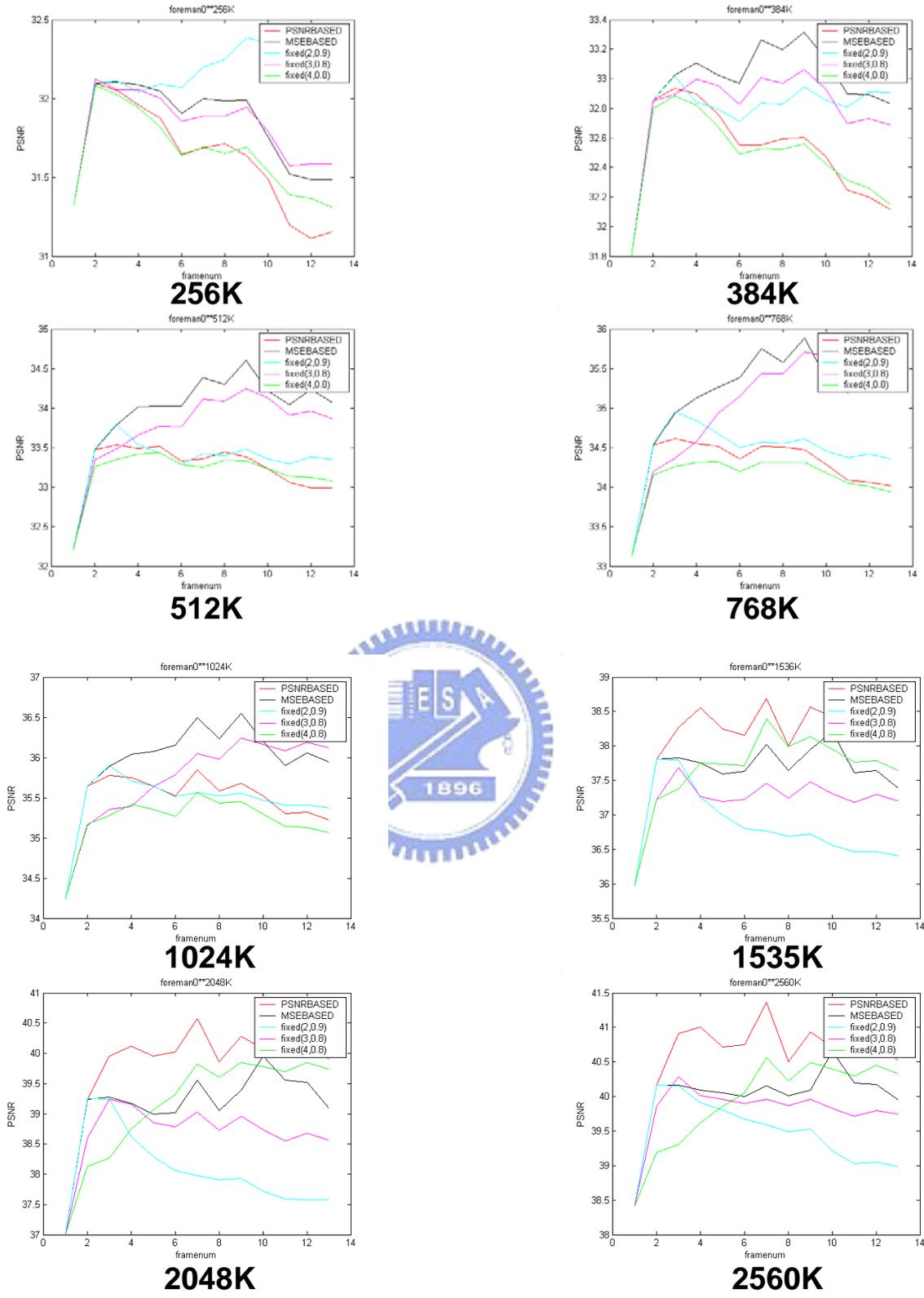


圖 3.8 foreman frame-based 結果

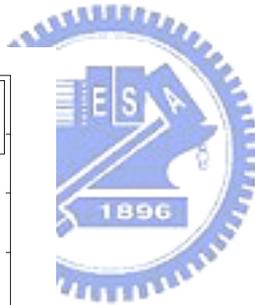
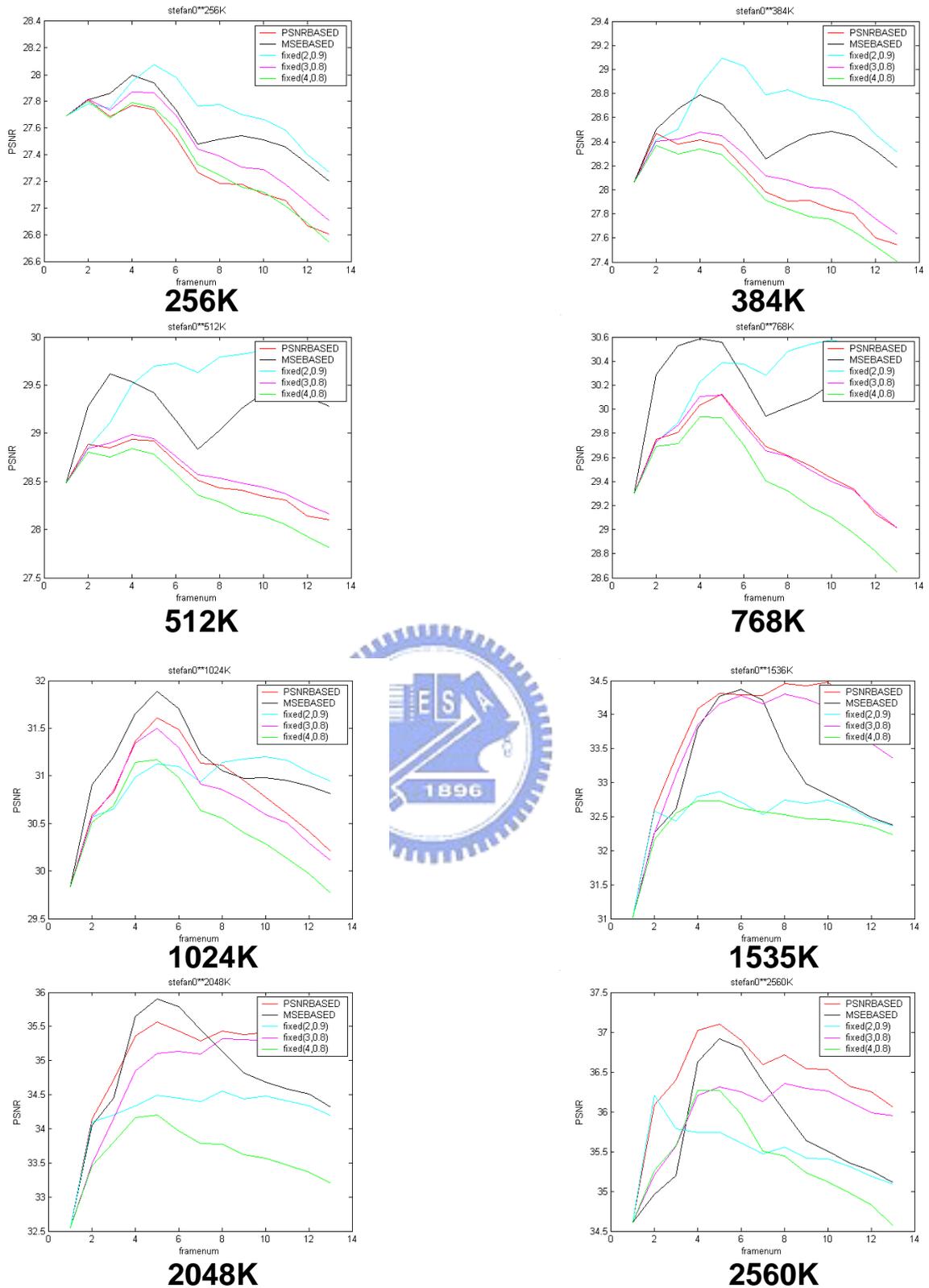
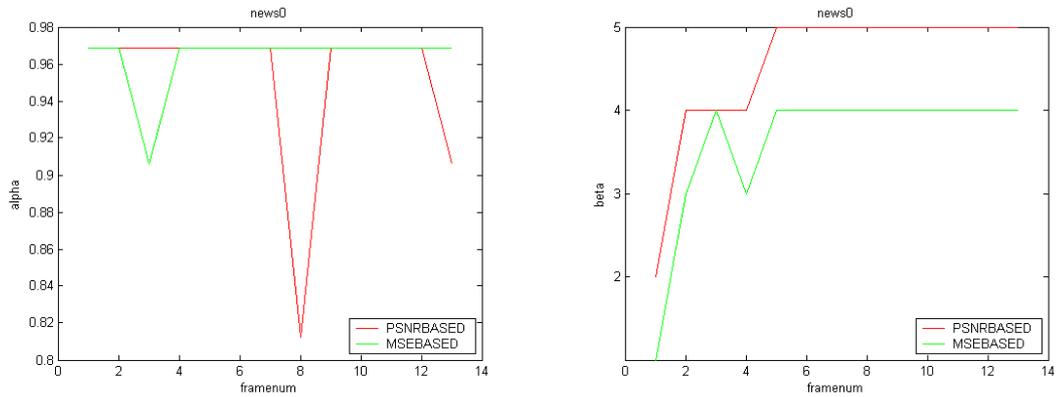


圖 3.9 stefan frame-based 結果

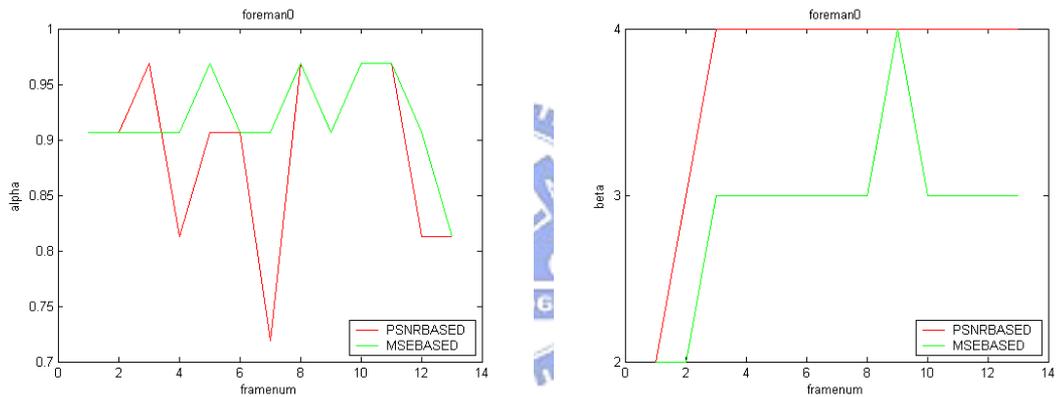
由以上的結果我們可以發現，在比較差的頻寬條件下，使用 MSE 為基準的最佳參數挑選模式，將可以得到非常明顯的影像品質改善，而在比較好的頻寬條件下，所得到的品

質下降則是非常有限的，這跟上節中，對於 MSE 和 PSNR 的特性比較，有著相當程度的吻合。

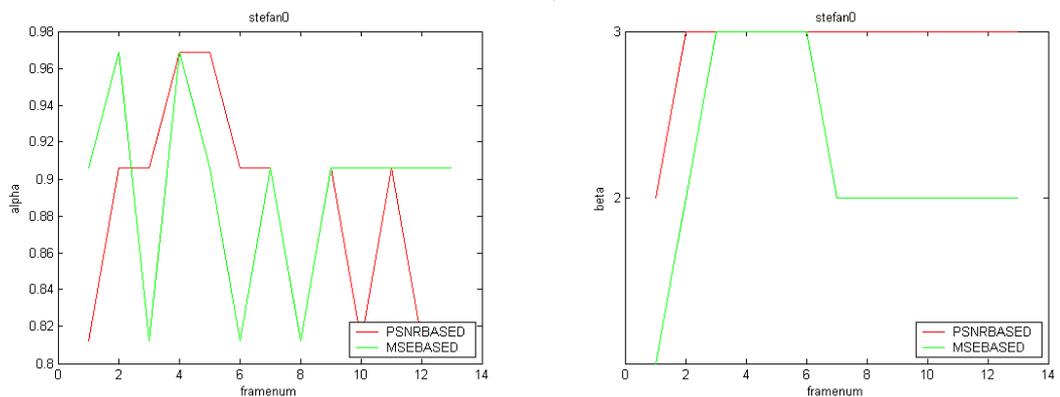
下面的圖則是將兩種不同的方法所挑選出來的最佳參數 α β ，以圖型的方式，作一個比較：



(A)



(B)



(C)

圖 3.10 MSE 方法和 PSNR 方法最佳參數比較圖
(A)(B)(C)分別為 News Foreman Stefan 的結果

明顯地，我們可以發現在以 MSE 為標準的方法中，考量低頻寬時的情形較為平等，

所以挑選出來的 β 值較小，改善了在低頻寬時過度嚴重的錯誤漂移[error drift]現象；而以 PSNR 為標準的原始方法中，因為太過於重視高頻寬時的情形，所以挑選出來的 β 值普遍較高。而在另一個參數 α 的挑選上，則兩個方法大致相同，雖然有一些不一樣，不過沒有很明顯的規律，不過都挑取了相當高的 α ，可見使用較高的 α 來提升編碼效率，是比較好的方法。

下面的圖我們將表示出在所有的檢查點的平均 MSE 和 PSNR 值:

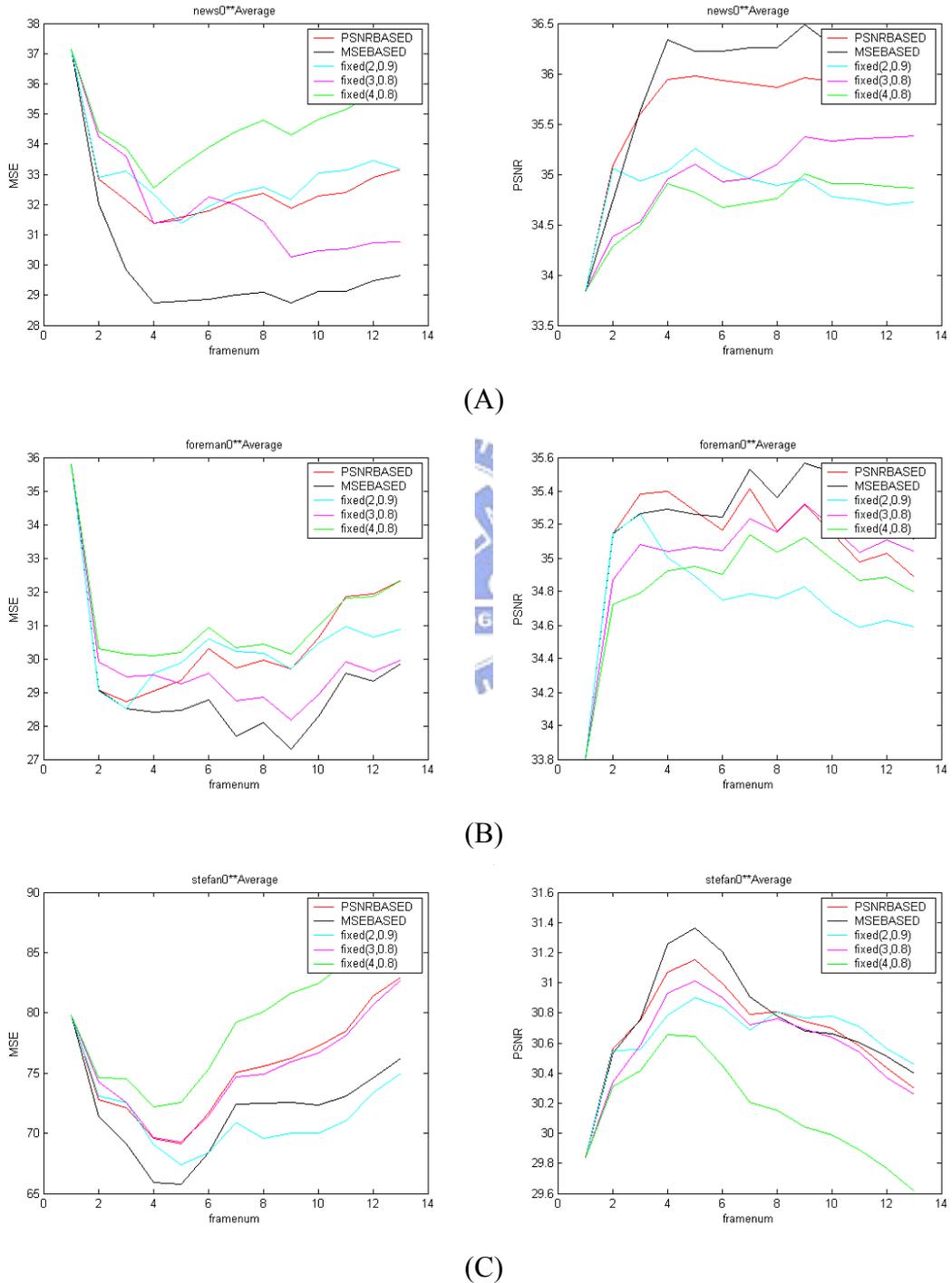


圖 3.11 平均 MSE PSNR 結果比較圖
(A)(B)(C)分別為 News Foreman Stefan 的結果

另外，我們也發現，在這個挑選的過程中，因為只考量本張的參數組合對於下一張影像的影響，而不是考量整串影像的情況，所以對於 error drift 的估算遠較實際情形來得輕微，很容易在一開始就挑選了比較不適合的參數，而在後續的挑選過程中，因為前面的錯誤，而無法修正回來。在我們的測試結果中，以 MSE 為標準的方法，比較注重在低頻寬時的效果，也可以說是比較注重 error drift 的情形；而以 PSNR 為標準的方法中，則因為 PSNR 的特性，所以過份的注重在高頻寬時的效果，所以導致在低頻寬時，產生了比較嚴重的 error drift，嚴重影響整體的影像品質。

在我們的結果示意圖中，可以發現，因為 error drift 的關係，使得就算以平均的 PSNR 作為比較標準，PSNR 方法所得到的效果，也只有在一組影像中的前幾張比 MSE 方法所得到的效果好，之後都比較差，如果以 MSE 作為比較標準的話，則是全部都比 MSE 方法來得差。由這樣的結果，我們也了解到了 error drift 對於編碼效果影響的嚴重性，在之後的估算中，將特別地注重此部份的研究。

3.4 總結

在本章中除了介紹原始的參數 α β 最佳化方法(採用 PSNR)，也提出了為了平等的考量高低頻寬的重要性，而改成採用 MSE 的方法。經過改善後，大幅度的提升了在低頻寬時的效果，而在高頻寬時的效果降低相當有限，所以在我們接下來的論文發展中，也將以 MSE 的大小作為決策的標準，最後在比較不同作法的整體效能時，再使用 PSNR 為衡量的依據。下圖為本方法的流程示意圖：

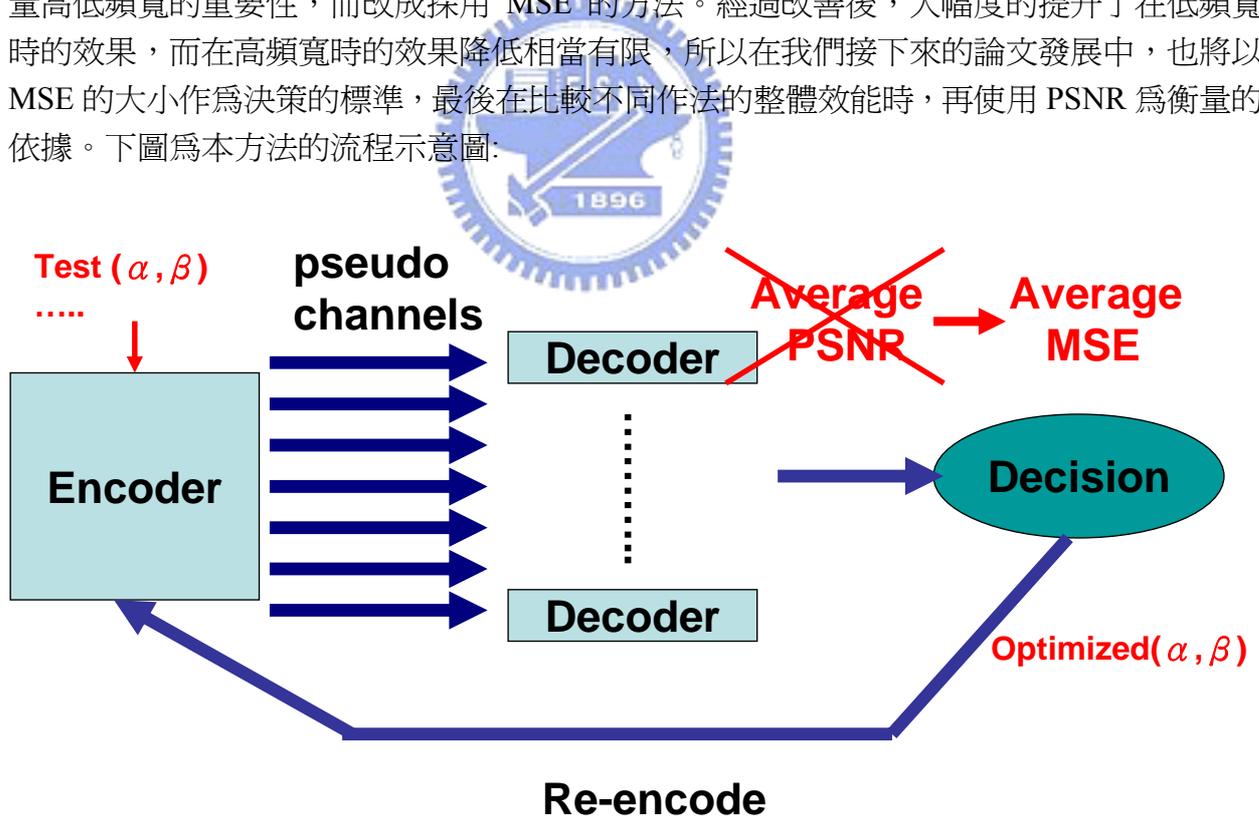


圖 3.12 最佳化流程示意圖

由上圖中我們可以了解到，採用這個方法來搜尋最佳化參數，需要經過編碼端、通道、解碼端，在計算 PSNR 或是 MSE 之後，才能經過一個比較機制來得到最適合的參數

組合。這樣的作法需要非常多時間的，在目前的電腦環境中，編碼一張影像需要超過三個小時才能完成，這樣的速度就算是離線作業，也是不可行的。另外，由之前的分析中，我們也了解到本方法中，只考量到參數組合對於下一張畫面的影響，並不是考慮接下來一整串畫面，所以當錯誤發生時，會一定程度地低估了錯誤處罰的嚴重性，而影響最佳參數組合挑選的準確性。

雖然本方法在速度上和錯誤漂移的考量上有很大的缺失，不過仍然証明了對於不同的影像，不同的頻寬條件下，對於不一樣的畫面，甚至是同一張畫面中的不同 MB，依照不同的影像選取不同的參數組合，可以帶來不同程度的品質改善。



第四章 數據分析

在第三章中所介紹的方法，驗證了我們原先的假設，就是依照不同的影像特質，對於不同的 FRAME，甚至是不同的 MB，個別採用其適合的最佳參數組合，將可以使得編碼後的影像，在一整段的可能頻寬間，得到一個最好的平均效果。但是在另一方面，因為原本搜尋的方法，不論是 FRAME BASED 或者是 MB BASED，雖然已經採用了三階段搜尋法來縮小尋找的範圍，需要測試的組合還是非常的多，而且前述的方法雖然經過加速，不過還是一種測試法，包含編碼，通過假設性的通道(channel)，再解碼，計算最佳的參數，如果要取得 MB BASED 的最佳參數組合的話，在現有的電腦環境下，一張 cif 大小的影像，將需要超過三個小時才能夠編碼完成，就算編碼的須求是事先完成(off line)，但是在時間上還是近乎不可行。

另一方面，在原本的方法中，因為在挑選最佳參數的過程中，只考量到下一張的情形，大幅地低估了 error drift 的影響，很可能挑選出了不適合的參數組合，如果要改成考慮兩張，甚至多張的綜合情形，將使得原本就已經非常慢的運算速度，雪上加霜。就算改成使用了我們所新提出的以 MSE 為考量基準的方法，雖然在高低頻寬的比重上，解決了先前計算的盲點，相當程度地改善了整體的影像品質，在 error drift 的考量方面，也因為加強了低頻寬重要性，所以也改善了部份低估 error drift 的缺點，但是還是有相當大的改善空間，在接下來我們所發展的快速猜測法中，將對於 error drift 的影響，完整地來列入計算。

有鑑於此，利用了前述的方法驗證了採用不同的參數組合有其價值之後，我們將試著發展出一套猜測的方法，利用影像不同的特質，例如 motion vector 大小、影像複雜度、residual 大小等等，輔以建立的模型(module)，在經過簡單的估算後，就可以挑選出接近於最佳化的參數組合，希望發展出來的方法，可以在稍微地增加運算量的情況下，改善相較於固定參數組合下的影像品質，並且在可能的情況下，盡量逼近前述方法的效果。

在本章中，我們將試著利用一些方法，統計分析在不同條件下影像的特性，希望在這些繁瑣的數據中，發現一些規律，建立成模型，提供快速的參數決策過程中來參考。在本論文的分析中，將以 FRAME BASED 的方式來進行，考量不同影像的一些特質。在前兩個小節中，我們將分別對於影像經過解碼端之後的特性進行分析，第一節中，我們將分析不同參數 β 之選擇，如何對於最後的影像效果帶來品質提升，也就是對於 MSE gain 的大小進行估測。而在第二節中，我們將試著去了解當通道狀況不佳時，產生的錯誤和伴隨而至的錯誤漂移[error drift]之特性，經過一些實驗的設計，找尋出一個可以快速的估算出不同頻寬條件下錯誤對於影像品質的影響大小，也就是對於 MSE lost 的大小進行預測。接下來在第三節中，為了進一步把參數的決策提前，我們對於編碼端內部的結果關聯性進行了解，提出了一個可以由 residual 大小來預測最後編碼完成後 bit-stream 大小的模型。最後，在第四節中，我們將作一個簡單的總結。

4.1 MSE gain

在原本的最佳化參數挑選方法中，要得到最佳參數組合在每次採用不同的參數組合進行測試時，都必須要經過完整的編碼、通道、解碼後，再對於得到影像計算 PSNR 或者是 MSE，最後再挑選最適合的參數組合，雖然較為精確，但是也使得所需耗用的時間過長，而幾乎不實用。因此，我們接下來將分析影像在不同的頻寬下，經過解碼端後，會產生怎樣的影像品質變化，希望能夠直接在編碼端，就可以預測經過通道和解碼後的 MSE 變化，簡化參數挑選的流程，想法如下圖所示：

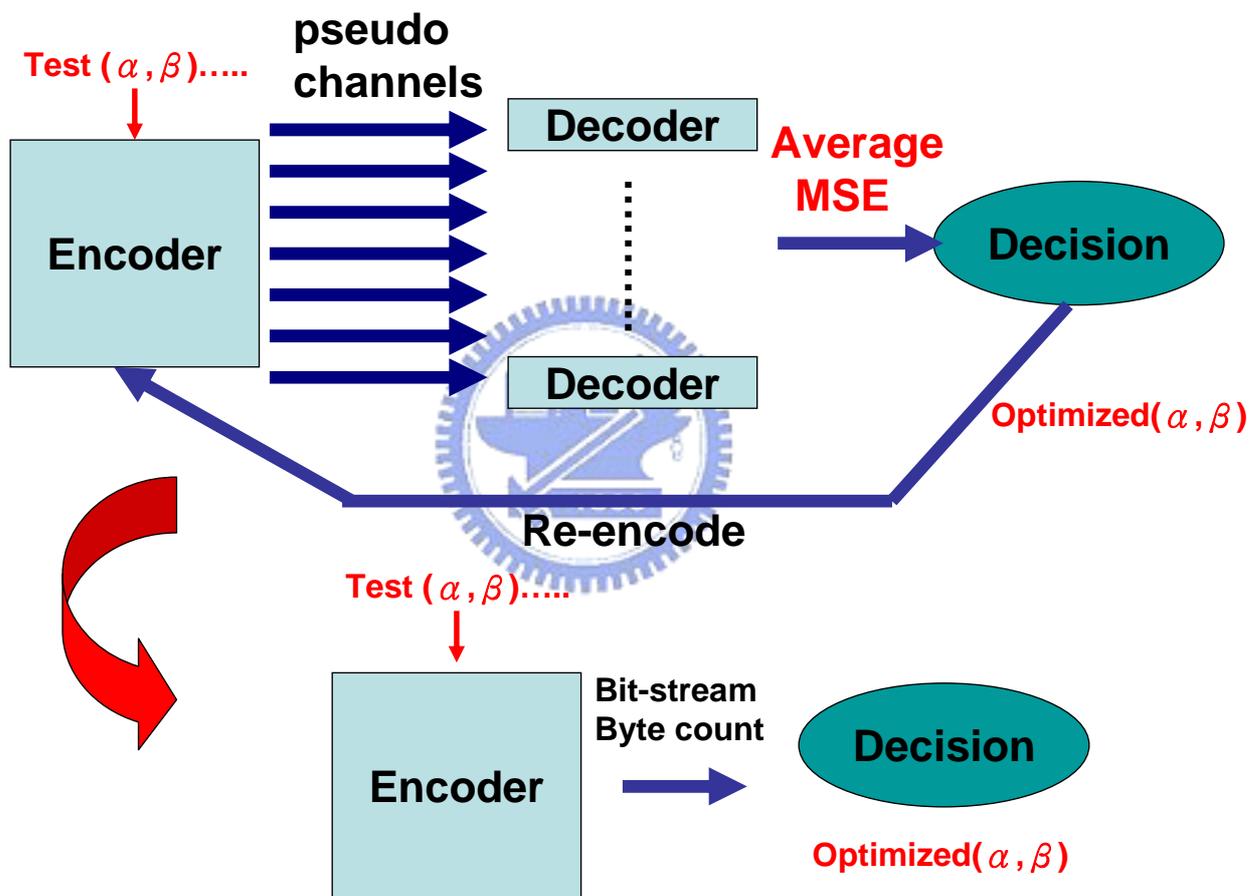


圖 4.1 MSE 估算示意圖

在接下來的兩節中，將分別對於參數如何降低 MSE 和錯誤如何增加 MSE 來進行分析，在。而在本節中，第一小節將討論 MSE gain 的成因來源；第二小節將對於 MSE gain 的大小進行分析統計；最後在第三小節中，我們將作一個總結，並依此提出規則二。

4.1.1 MSE gain 的來源

當我們使用不同的參數 β 來編碼時，將使運動補償的效率改變。當 β 值採用得較高時，將可以使得經過運動補償後的 residual 變得較小，最後編碼完成的 bit-stream 也將變得較短，byte 數較少，使得在相同的頻寬條件下，解碼端可以得到較多的資訊量，進而使得最後經過解碼後的影像品質提升。反之，當 β 值採用得較低時，將使得運動補償效率不佳，解碼端在相同的頻寬下，可以獲得的資訊變少，還原的影像品質較差。其想法如下圖所示：

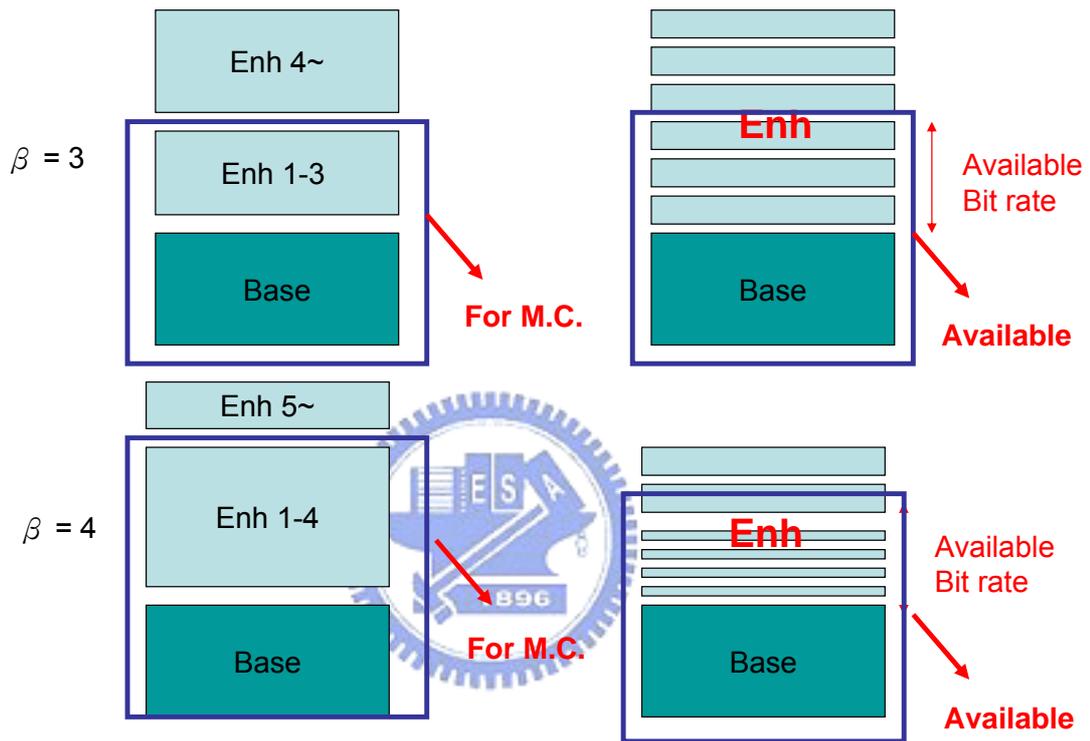


圖 4.2 MSE gain 來源分析

我們實際使用不同的參數組合對於影像進行編碼，而產生不一樣大小的 residual，下面的圖表示出在不同的 β 值時(α 設為最大值 1)，所得到的 residual Y 元素大小，其中顏色愈深的部份表示 residual 的值愈大，愈淺的部份則表示 residual 的值愈小，所採用的測試影像為 News。

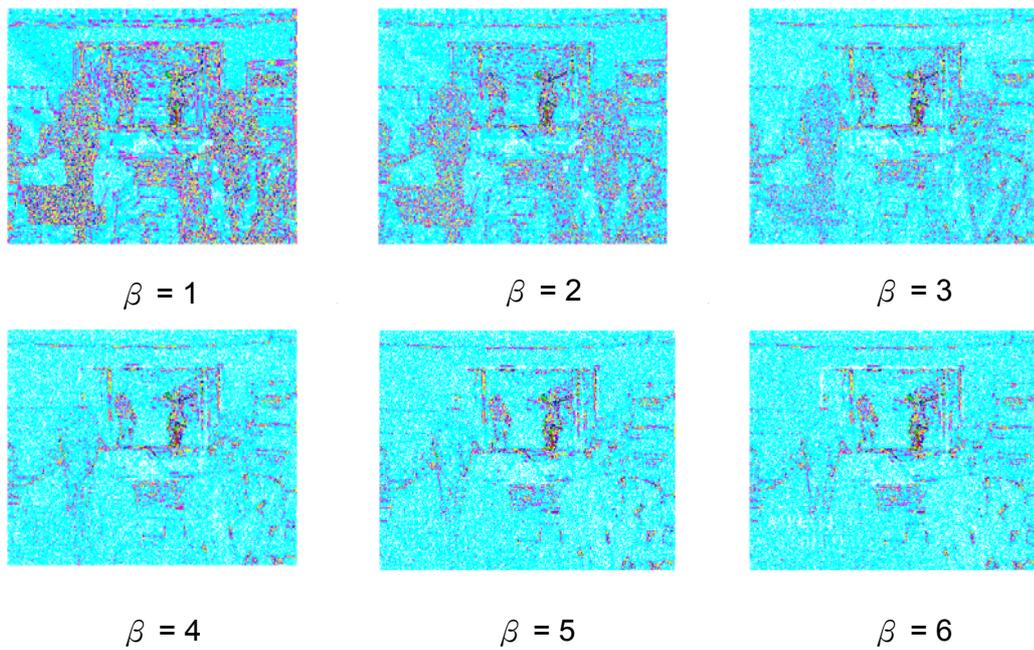


圖 4.3 Residual 大小示意圖

爲了更清楚地了解使用不同的 β 值時對於 residual 大小的改變量，我們把不同 β 值之下的 residual 差值用下圖表示出來。同樣地，顏色愈深的部份表示差值愈大，採用的測試影像也是 News。

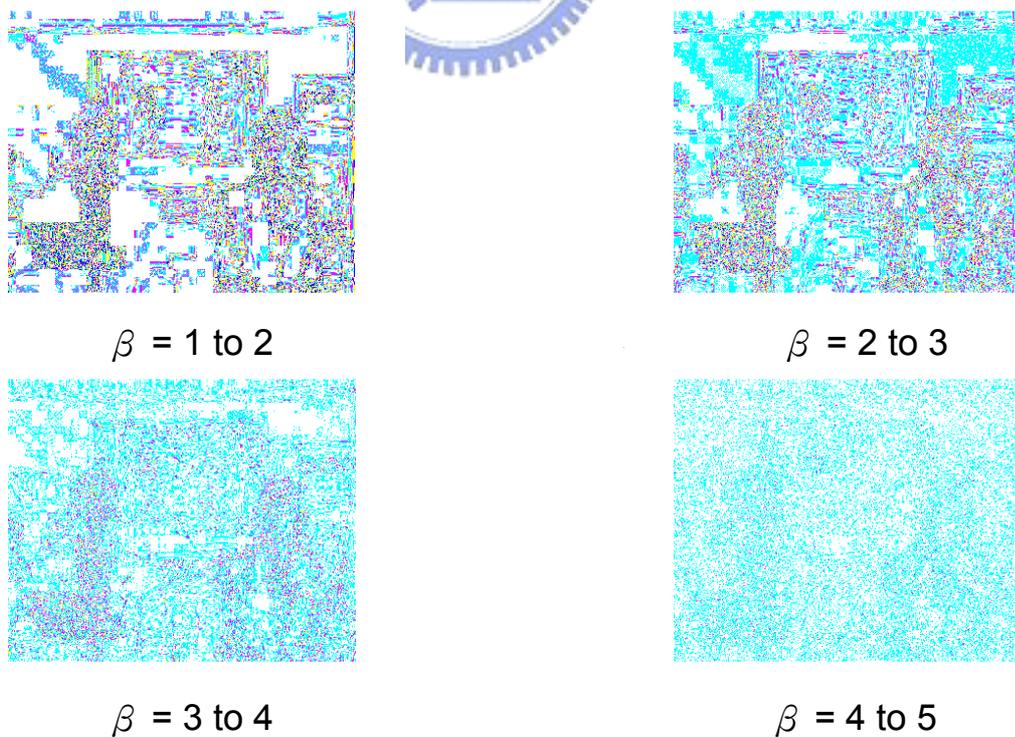


圖 4.4 Residual 大小差值示意圖

由以上的結果可以明顯地看出 Residual 大小和所採用的 β 值有很大的關聯性，愈大的 β 可以使得最後的 residual 愈小，進一步地使得編碼出來的 bit-stream 長度較小，得到較好的編碼效率。也可以發現愈低層的加強層資訊被採用時，可以減小的 residual 大小愈大相對，比較高的加強層資被採用時，所帶來的好處就較為有限，這個特性在最後的 β 值選擇時，將被考慮。

4.1.2 MSE gain 分析

在了解了不同的 β 值如何影響編碼出來的 bit-stream byte 數後，爲了進一步了解 byte 數和解碼端所得到的影像品質(MSE)之關係，我們實作了一個特殊的通道，可以控制其傳給解碼端的 byte 數。我們測試了使用不同的 β 值，接收到不同的 byte 數(0~20000)，每 200 個 byte 作一次統計，以 News 爲例，得到了下圖中的結果，發現 MSE 大小和 byte 數，在不同的 β 被使用時，都是呈現一個由數段線性所組成的關係，而且，在不同的 β 曲線中，都有斜率非常相近的線段，只是位置和長度大小不同而已。另一方面，使用不同的影像進行測試時，所得到的關係都是類似的。

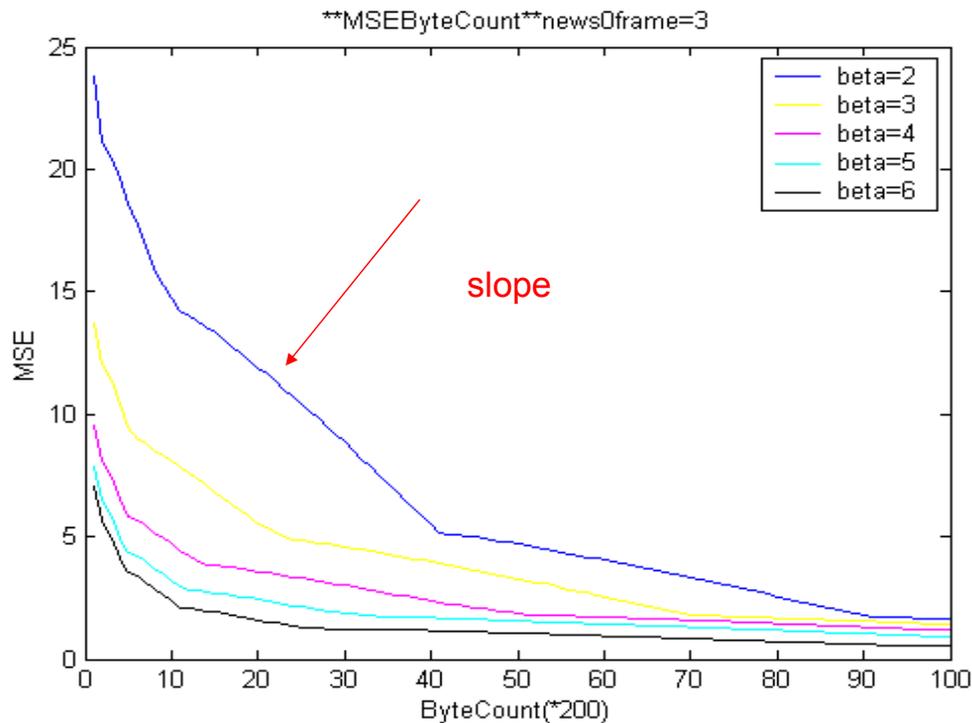


圖 4.5 MSE 和 Byte 數關係圖

接下來，我們採用描點的方式，把不同 β 值，不同畫面的結果，畫成如下的示意圖。可以清楚地發現，同一層的資料量(byte 數)對於最後的 MSE 貢獻是一致的，所以我們只要知道每一層資料量的大小(bit-stream 分佈)和頻寬大小，就可以不需經由實際的解碼動作，就估算出解碼之後效果的改變情形。

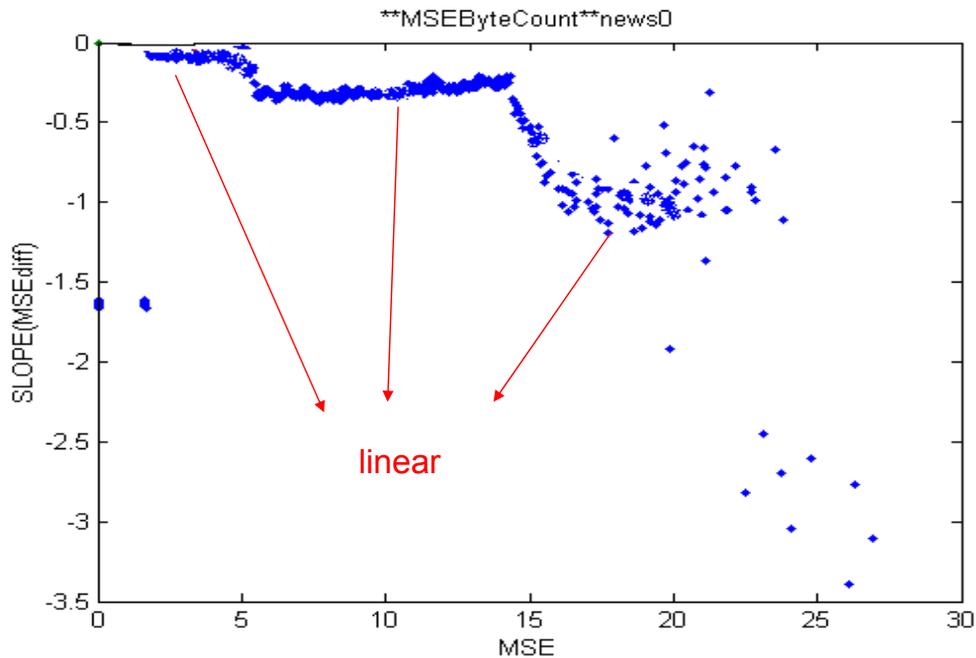


圖 4.6 MSE gain 和 MSE 值關係圖

4.1.3 總結(規則一)

經由測試不同的影像，我們發現這些線性的關係也是通用的，唯一的不同是，因為不同影像 residual 的特質，會使得同一層的資料量大小不同。另外，MSE 計算的基準，將隨著不同的影像而有所不同，但是同一層資料對於最後 MSE 大小的貢獻情形是一樣的。我們把所得到的結果整理於下表：

基本層和第一層加強層的資料量為基準值	
Bit-plane 2 每 200bytes 對於 MSE 之貢獻	3.5
Bit-plane 3 每 200bytes 對於 MSE 之貢獻	1
Bit-plane 4 每 200bytes 對於 MSE 之貢獻	0.3
Bit-plane 5 每 200bytes 對於 MSE 之貢獻	0.1
Bit-plane 6 每 200bytes 對於 MSE 之貢獻	0.05

表 4.1 MSE gain 和 byte 數關係表

因為第一層資料量過小，而且其對於 MSE 大小的貢獻情形較為不一致，而且在頻寬不要太差的前提下，第一層的資料大多都會順利地傳到解碼端，所以不會影響到 MSE 的基準，所以我們也不予考慮。

由上表中的關係，我們稱之為規則一，可以在頻寬已經知道的情況下，約略地估計出經由實際的解碼後最後的效果改變情形。

4.2 MSE lost

在我們採用了部份加強層的資訊來提升運動補償效率的同時，如果使用了過多的資料量，當頻寬不足時，將會對於畫面的效果帶來一定程度的影響，而且因為錯誤漂移的關係，將不只會影響到一張畫面，而是對於之後所有的畫面(到下一張 Intra 畫面為止)都有影響。如同第三章中的討論，錯誤漂移和錯誤處罰將要被完整地考慮，避免因為低估它的影響性，而降低整體的編碼效果。

在本節接下來的篇幅中，接在第一小節對於錯誤漂移[error drift]進行討論。在第二、三小節中分別討論錯誤漂移的兩項特性:可加性和指數型遞減性。在第四小節中將討論錯誤處罰對於 MSE 大小的影響。最後，在第五小節中，將作一個總結，並且整理出關於 MSE lost 的兩項規則: 規則二和三。

4.2.1 錯誤漂移(error drift)

在影像壓縮編碼中，達到好的壓縮效果主要的貢獻就是來自於運動補償的採用，也就是對於接連的兩張畫面中，只編碼其有差異的地方(residual)。因為一般的情形之下，接續的兩張畫面會十分地相近，因此需要編碼的部份變成非常的小，藉此達到良好的壓縮效果。

但是相對地，如果前一張的資訊因為頻寬不足，或者是網路問題而無法被解碼端完整地接收時，如果這些遺失的部份沒有被運動補償所採用，則只會對於該張畫面帶來影響。但是如果遺失的部份被運動補償作為參考時，因為在作編解碼時，已經把之前一張畫面中遺失的資料假設為收到，所以這部份的資料並沒有被 residual 所包含，不論傳送下一張的頻寬為何，都無法補回，這種錯誤會對接下來的一連串畫面品質帶來影響，這樣的影響會一直持續到下一張 I (Intra)畫面為止。如下圖所示:

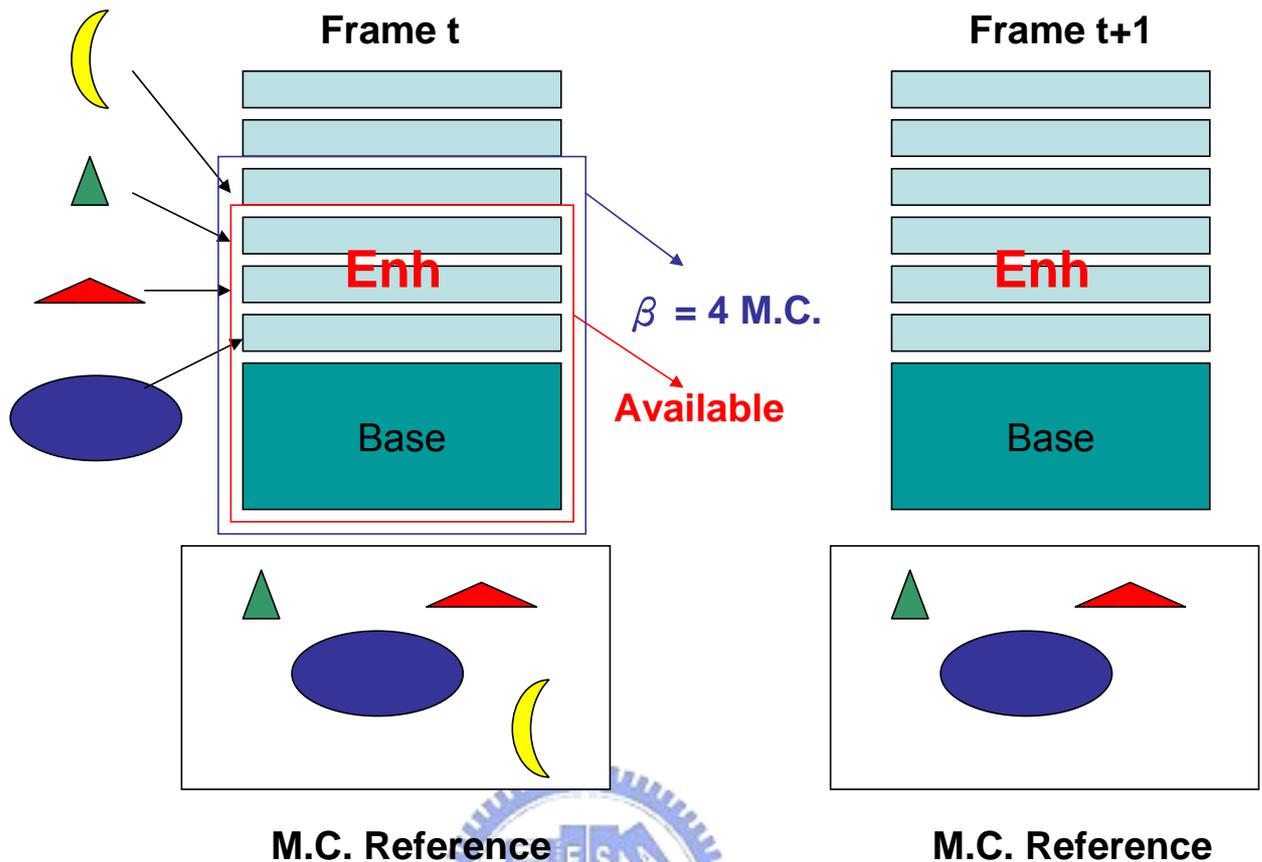


圖 4.7 error drift 示意圖

在接下來的幾個小節中，我們將設計一些實驗，來了解 error drift 的特性，希望可以建立起相對應的模型，以達到可以在編碼時就預測 error drift 大小，而對於參數組合的選擇作調整的功能。

4.2.2 可加性

在實際的網路傳輸中，通常當頻寬不足時，不會只維持一張畫面，而很可能是一長串的畫面都有頻寬不足的問題。爲了了解當接續幾張畫面都有錯誤處罰問題時如何整體地考量，我們設計了實驗來對於此類的問題進行分析。在前兩次實驗中，我們分別讓頻寬不足的問題在第一和第二張畫面發生，然後在解碼端量測對於第三張畫面的影響，之後再把兩都對 MSE 造成的增加加總。在第三次實驗中，我們讓頻寬不足的情形同時在第一和第二張畫面產生，接著在解碼端量測對於第三張畫面的影響。如預期地，我們發現兩種不同情況下，對於 MSE 的增加量是非常相近的，其實驗示意圖和結果示意圖如下所示：

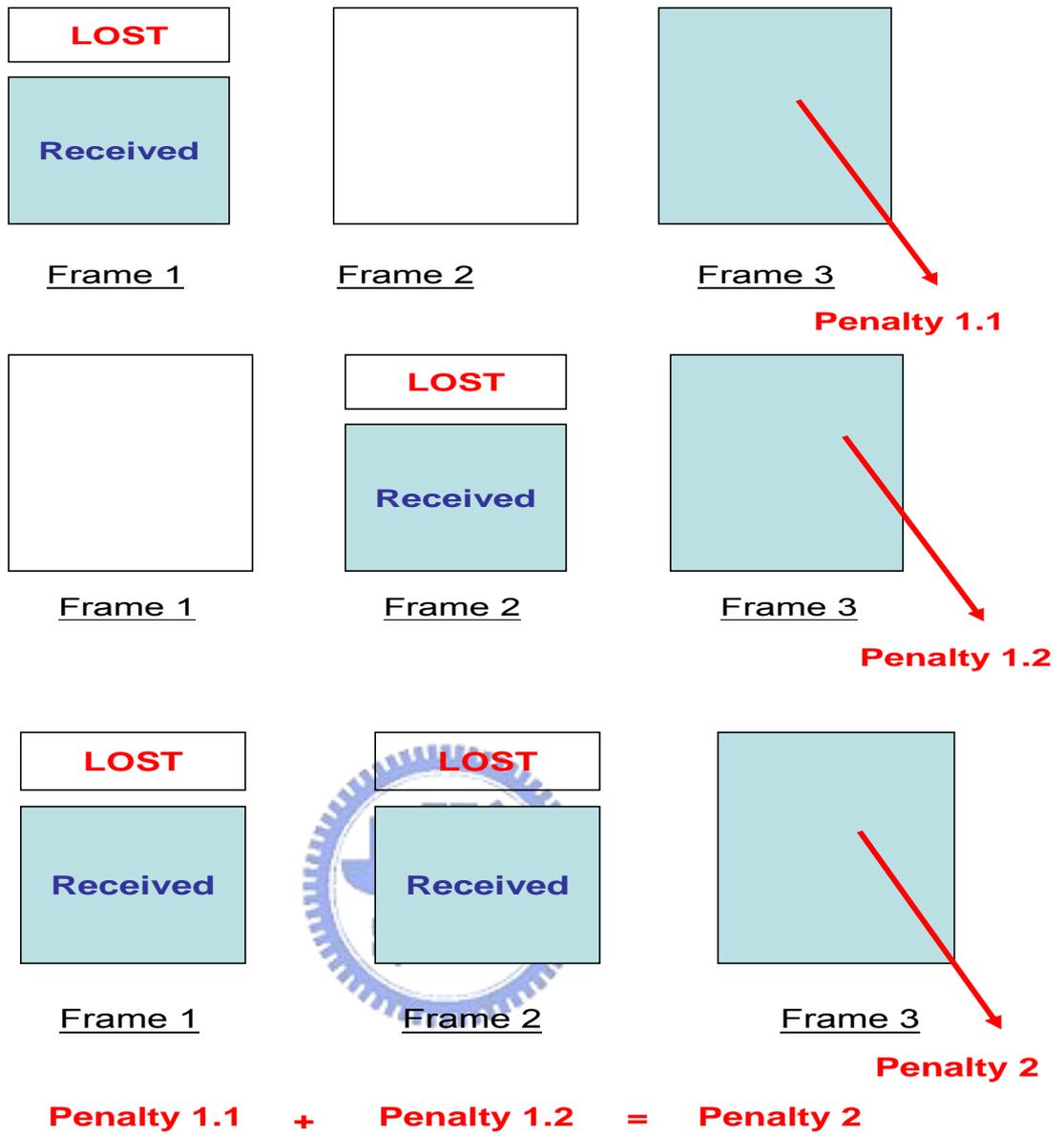


圖 4.8 可加性實驗示意圖

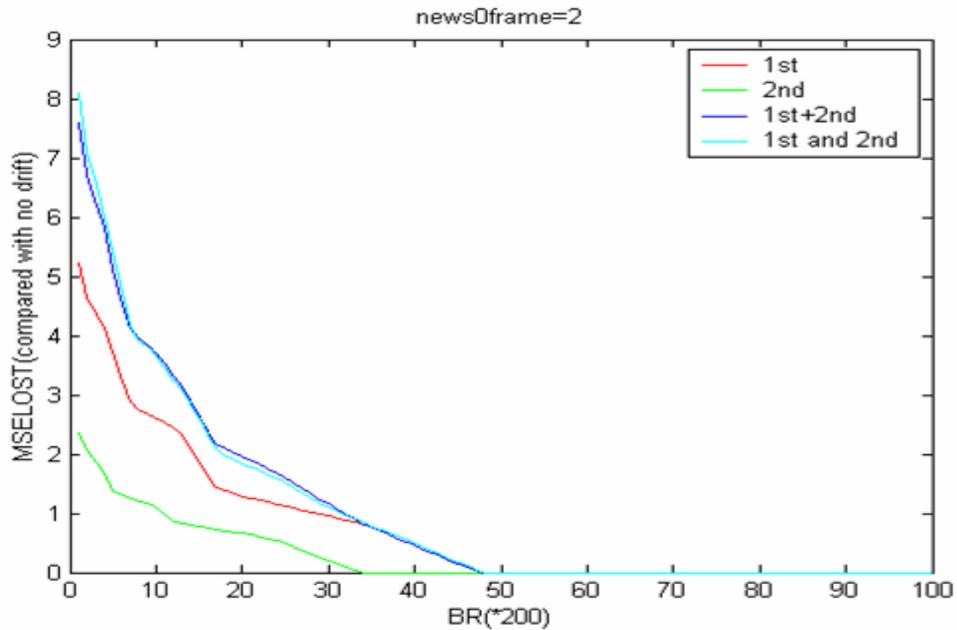


圖 4.9 可加性實驗結果示意圖

我們對於不同影像的不同畫面，進行了多次類似的實驗，都發現了類似的結果，所以驗證了錯誤漂移的可加性。也就是說，我們可以對於不同畫面所帶來的錯誤處罰個別地考慮(MSE lost)，最後再進行加總。

4.2.3 指數型遞減性

如同 RFGS 論文中所提及的，因為 α 的採用，一個錯誤處罰將在一連串的畫面中，帶來類似於等比級數的不同程度影響，我們也對於這樣的方法進行驗證。在 β 值固定之下，採用不同的 α 值進行編碼，然後讓運動補償時所需要的部份資訊在經過模擬通道後遺失，然後在解碼端了解錯誤對於接下來幾張畫面品質之影響，其結果如下所示：

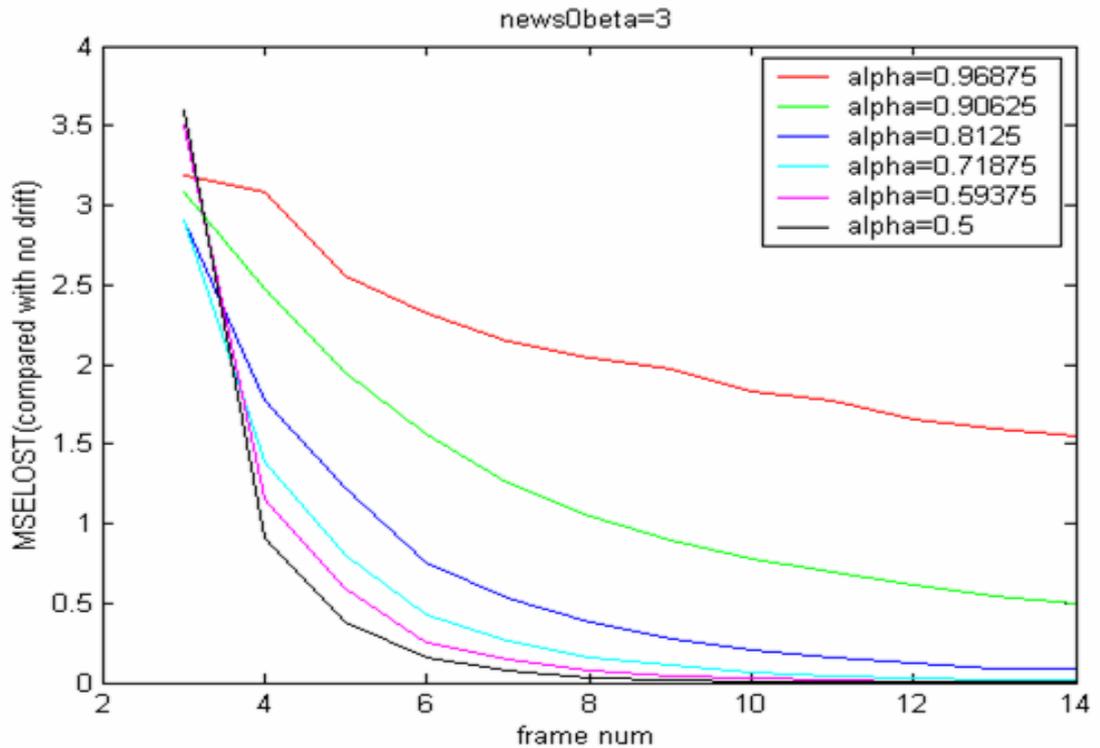


圖 4.10 指數型遞減性實驗結果示意圖

我們使用了不同的參數 β ，並且在不同的影像中進行測試，也使得通道遺失不同量的資料，發現都有類似於等比級數的特質，藉此我們也驗證了錯誤漂移的指數型遞減性。

4.3.4 MSE lost

類似於 MSE gain 的分析方式，為了達到在已知頻寬的狀況下，在編碼端就可以對於錯誤漂移對 MSE lost 的情形作一個估計，我們進行了以下的分析。首先我們先對於不同的影像，採用相同的 α 值，在不同的 β 值之下，觀察 MSE lost 和頻寬之間的關係，結果發現如下的結果：

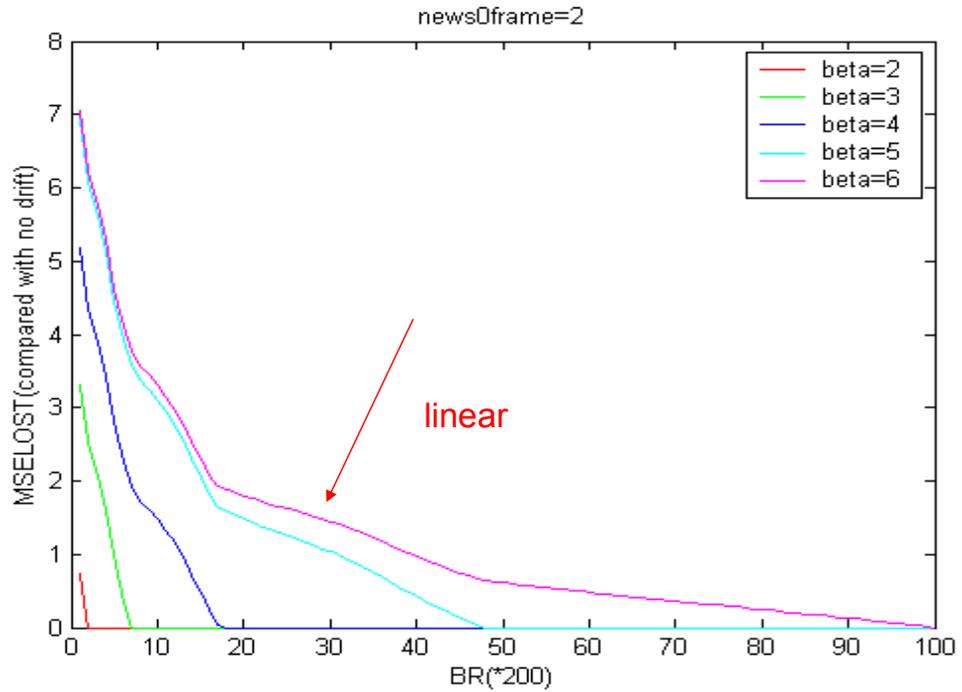


圖 4.11 error drift 大小示意圖

很明顯地，我們可以發現在上圖的結果中，MSE lost 的大小和頻寬的關係是呈現為很多段的線性關係所組成，而且在不同 β 值之下，似乎都有存在於相同斜率的線段，只是所在的位置和長度不同而已。和 MSE gain 的方法相似，我們把不同畫面，不同 β 值所產生的結果，用描點的方式，畫在同一張示意圖中。接下來的示意圖依序為三個不同的影像 News、Foreman 和 Stefan 的結果：

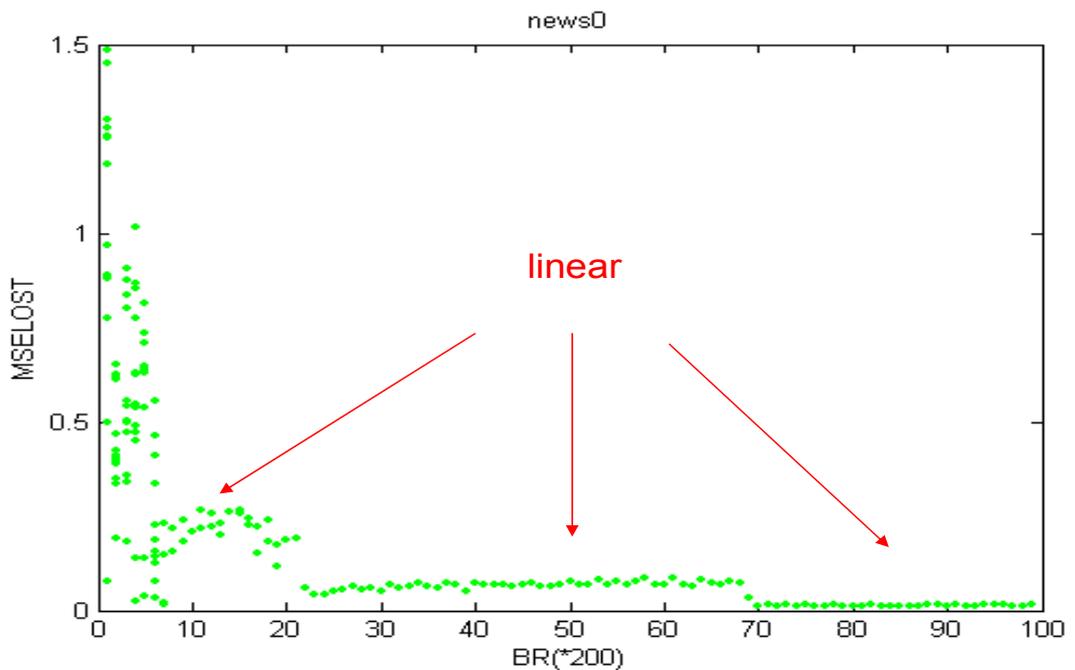


圖 4.12 MSE lost 和 byte 數關係圖(News)

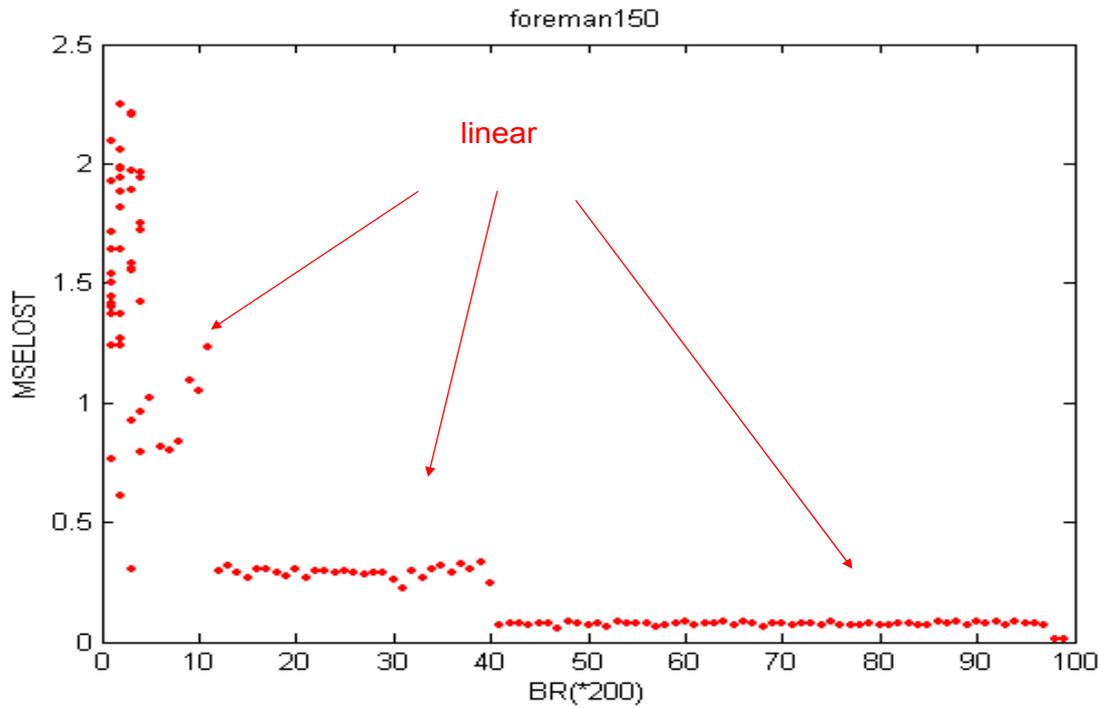


圖 4.13 MSE lost 和 byte 數關係圖(Foreman)

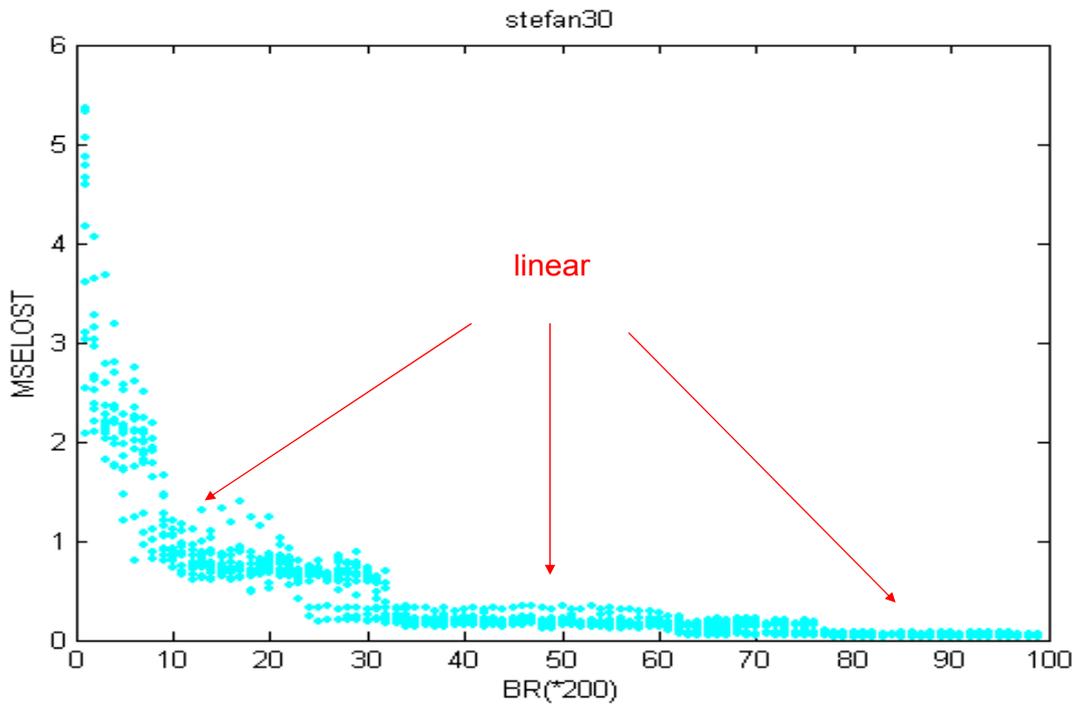


圖 4.14 MSE lost 和 byte 數關係圖(Stefan)

由上圖中，我們可以清楚地發現，在同一層的 bit-plane 中，相同大小的 byte 數的遺失，對於 MSE 的影響是一致的。而且我們對於不同的影像、不同的 β 值、不同的畫面進行實驗，可以發現關係是類似的，而且規律是一致的。

4.2.5 總結(規則二、三)

在進行了一到四小節的討論分析後，我們已經對錯誤漂移的特性有了一定程度的了解，爲了在編碼端對於 MSE lost 的大小進行估測，我們根據以上的分析，提出了兩個規則，分別是規則二和規則三。其中規則二是依據錯誤漂移的可加性和指數型遞減性建立，當了解到錯誤發生的位置和 residual 的大小以及可供使用的頻寬後，就可以對於這個錯誤所帶來的整體 MSE 上升，作一個約略的評估。其表示式和等比級數很相近，所示如下：

$$\text{Total MSE lost penalty} = \text{MSE lost next frame} * (1 - \alpha^n) / (1 - \alpha)$$

另外，根據對 MSE lost 的分析，可以得到一個類似於 MSE gain 的表格，我們稱之爲規則三，可以對於當錯誤發生時，約略估計出對於下一張畫面 MSE 的增加量，所示如下：

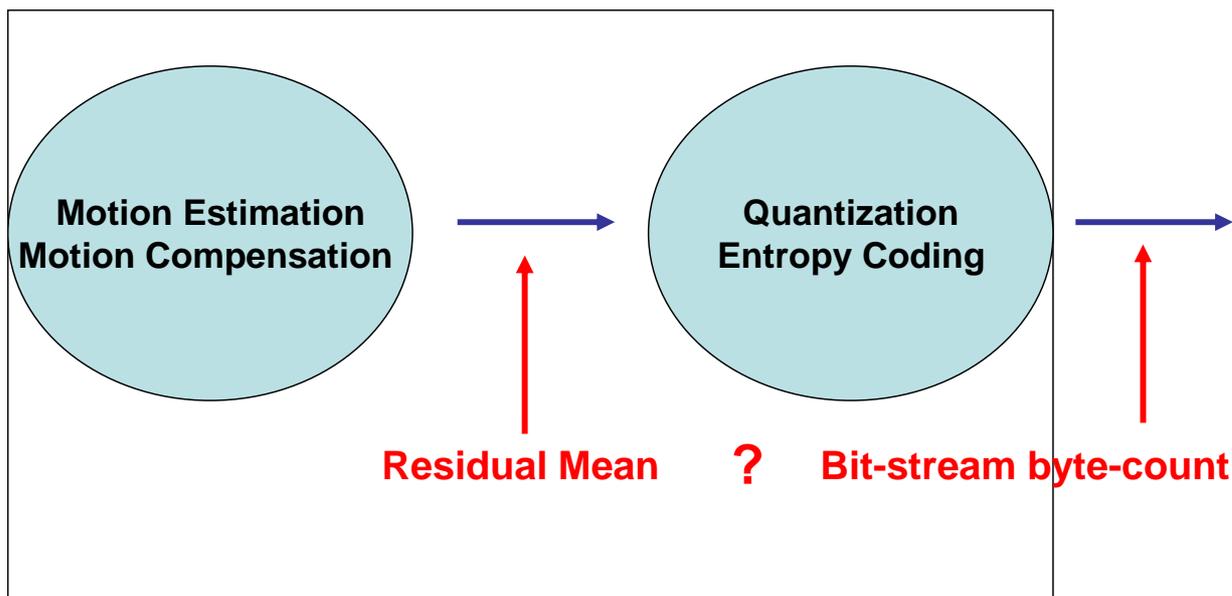
基本層和第一層加強層的資料量爲基準值	
Bit-plane 2 每 200bytes 對於 MSE 之增加	3.5
Bit-plane 3 每 200bytes 對於 MSE 之增加	1
Bit-plane 4 每 200bytes 對於 MSE 之增加	0.3
Bit-plane 5 每 200bytes 對於 MSE 之增加	0.1
Bit-plane 6 每 200bytes 對於 MSE 之增加	0.05

表 4.2 MSE lost 和 byte 數關係表

有了上述的規則二和三之後，我們就可以在編碼端，對於可能發生的錯誤漂移情形作一個概略的估計，挑選出最適合的參數組合。

4.3 Residual 的特性

在編碼端中，主要的工作就是把原始影像，經過種種的處理，把剩餘無法省略的部份 residual，編碼成最後的輸出 bit-stream。而這些動作中，可以大致分解成兩個部份。如下圖所示，第一部份主要包含了 motion estimation 和 motion compensation，最後的輸出是 y、u、v 三部份的 residual。第二部份則是把 residual 經過量化後，再用 entropy 編碼的方式，產生出最後的編碼結果。在第一小節中，我們將分析第一部份的輸出，residual 的 y 元素(主要)，對其平均值作討論。第二小節中將對第二部份的輸出，bit-stream 的 byte 數作分析。最後在第三小節中，對以上的分析結果作一個整合，建立一個線性模型。



Encoder

圖 4.15 Encoder 示意圖

4.3.1 Residual 平均值

爲了方便和最後編碼出來的 bit-stream 大小作比較，我們將把原始的 residual 值，取用 y 元素的部份，把整張畫面中所有點的值作平均的動作，當作一個衡量的依據。使用不同的測試影像時，將會發現 residual 平均值有明顯的差距。另外，採用不同的 β 值，也會對於 residual 平均值有一定程度的影響。下圖中將表示出 residual mean 在 News 和 Stefan 中使用不同 β 值時的結果：

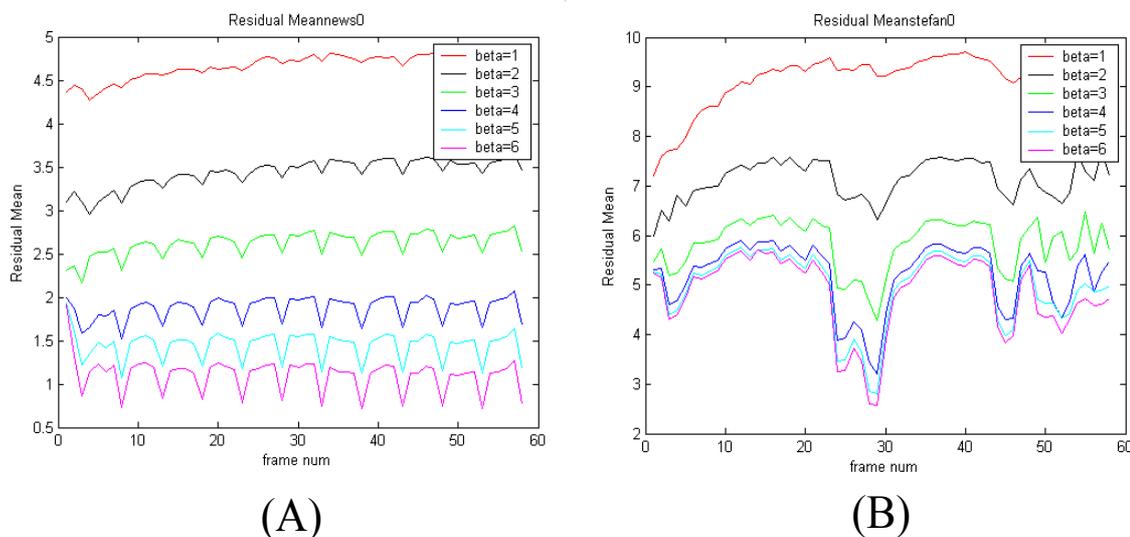
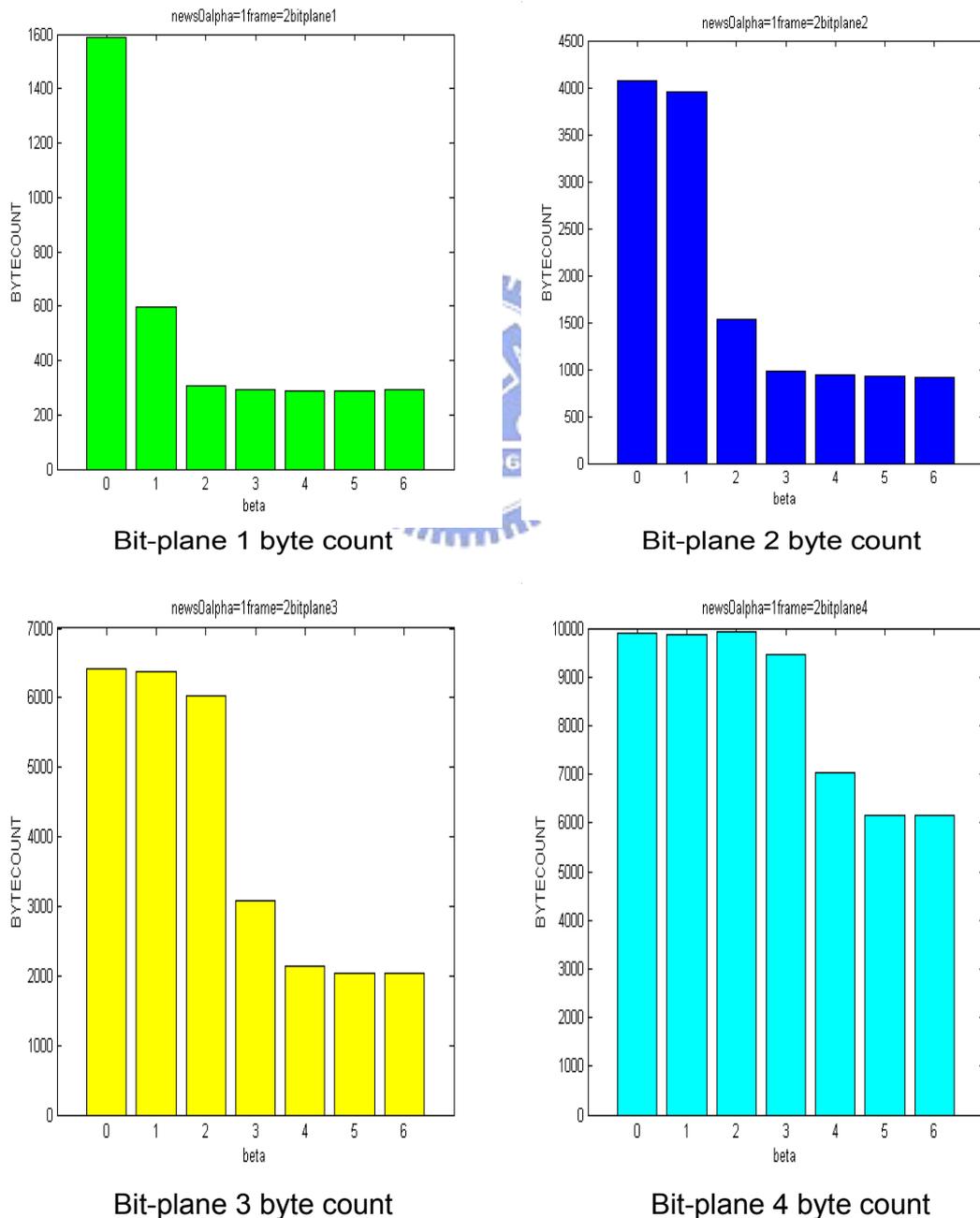


圖 4.16 Residual 平均值示意圖

4.3.2 Residual Byte 數

在 MPEG-4 的壓縮中，在經過運動估計[motion estimation]和運動補償後，還要經過量化和 entropy 編碼後，才會成為最後的輸出結果(bit-stream)。而在通道傳輸中，傳遞的就是 bit-stream。因為頻寬不足的關係，並不是所有的 bit-stream 都可以順利地被傳送到解碼端。MPEG-4 FGS 中採用的是位元平面編碼，資料量會依據其重要性，分佈在不同的位元平面中。在本節中，我們將對於 bit-stream 的長度大小，作統計和分析。

在 MPEG-4 的編碼中，在最後的輸出結果裡，會在每一張畫面的開始還有每一層資料的開始位置，加上一個註記，因此我們利用一個程式，可以統計出在不同的 β 值時，每一張畫面經過編碼後 bit-stream 大小的分佈，結果如下圖所示:



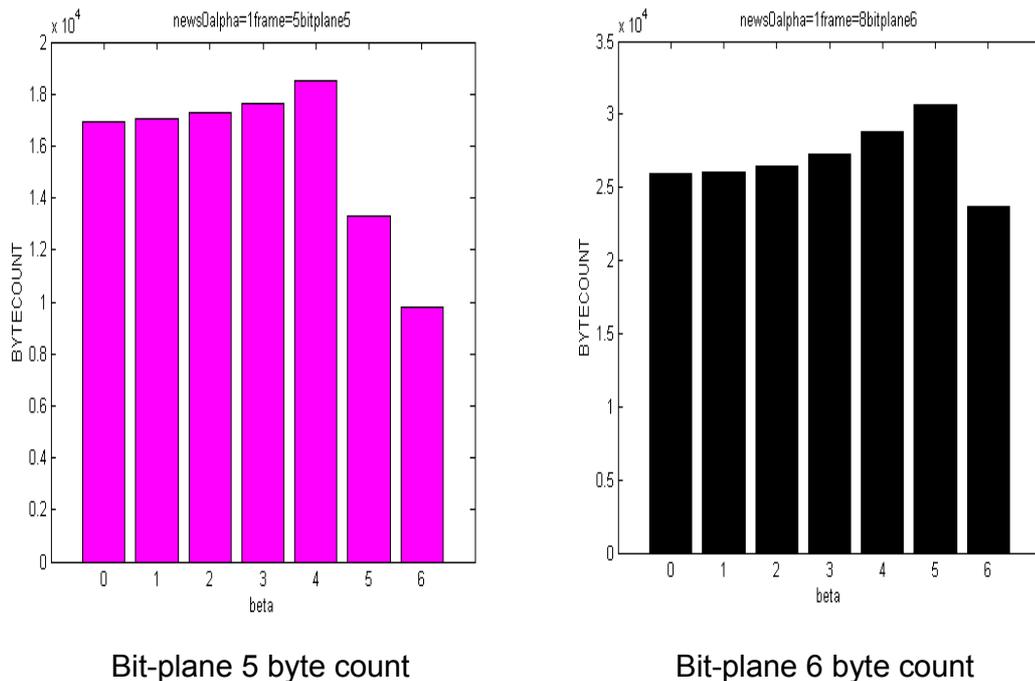


圖 4.17 Residual Byte 數示意圖

由上圖的結果可以發現，採用了愈大的 β 值，可以使得前 β 個 bit-plane 的 byte 數明顯地降低，達到提升編碼效率的效果。另一方面，比較重要的資料都是被安排在比較前面的 bit-plane，而且愈前面的 bit-plane 其 byte 數愈小，也符合位元平面編碼的主要精神，就是把大部份的資料，放置在最小可能大小的 bit-stream 中。

4.3.3 總結

在各自了解了 residual 平均值和 residual byte 數的大小關係後，我們認為兩個值的大小存在一定程度的關係。大致上就是 residual 平均值愈大，residual byte 數也會跟著增加，經過作圖比較後，發現到上述兩個數值將有一個幾乎是線性的關係，其結果如下圖所示：

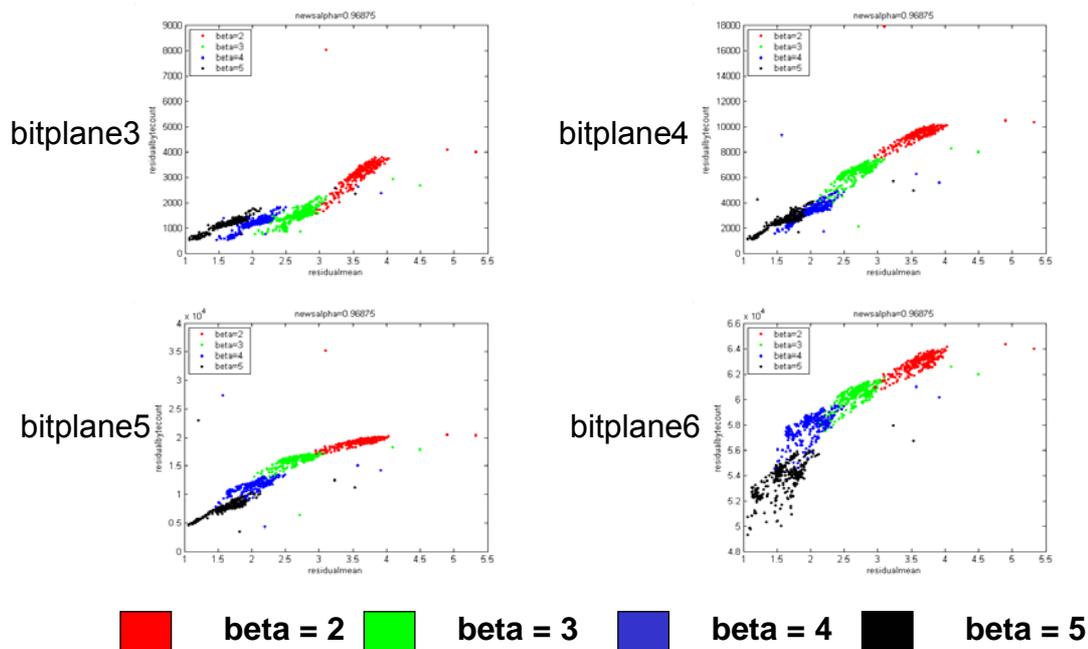


圖 4.18 Residual 線性模型(News)

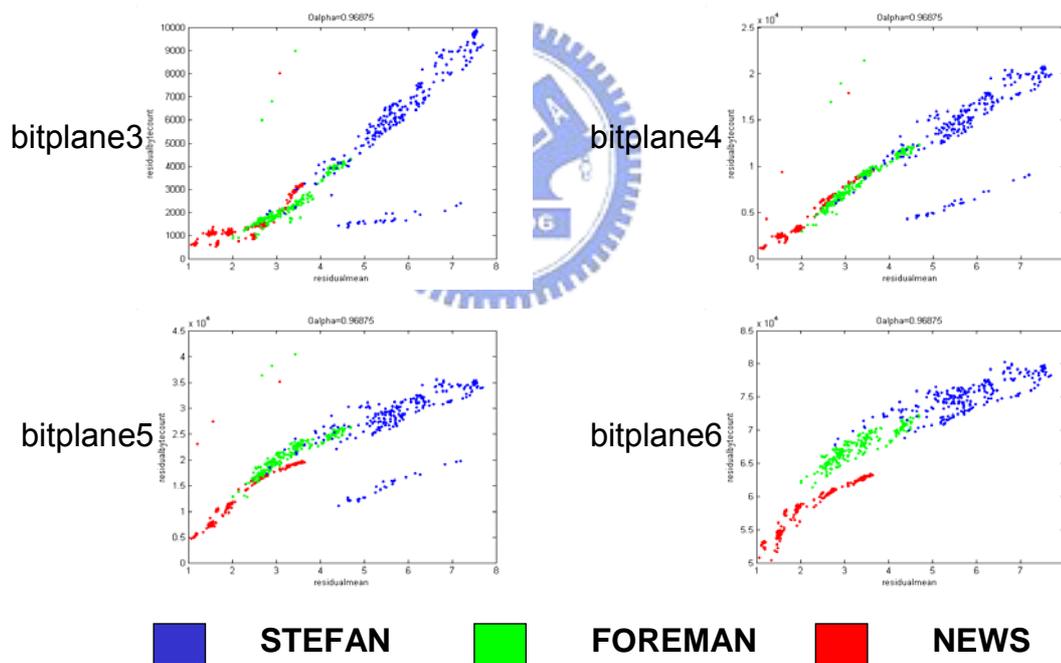


圖 4.19 Residual 線性模型(News Foreman Stefan)

如我們預期的在同一個影像中，不同張畫面，採用不同的 β 編碼，所得到相對應的 residual 平均值和 residual byte 數都會有接近於線性的關係。更進一步地，我們測試了三種不同的影像，快速的 Stefan、中速的 Foreman、以及慢速的 News，發現這個線性的關係是可以通用的。雖然會有部份的點偏離線性關係，但是已經提供了一個很好的估計方法。我們把發現的規則(規則一)，整理於下表中：

Residual 平均值 = M	
Bit-plane3 Byte 數	$M * 750 + 500$
Bit-plane4 Byte 數	$M * 2250 + 1000$
Bit-plane5 Byte 數	$M * 3750 + 5000$
Bit-plane6 Byte 數	$M * 3750 + 50000$

表 4.3 Residual 平均值和 Byte 數關係表

其中因為 bit-plane1 和 2 的 byte 數太小，所呈現的關係較為不規則，所以我們不予考慮。基本上，除非頻寬條件過差，否則 bit-plane1 和 2 的資料量應該都可以被順利的傳輸。

有了上述的規則以後，我們可以在編碼端經過簡單的運動估計和運動補償，就可以大致地了解最後編碼出來的 bit-stream 分佈，節省了一定程度的時間。而且，由實驗的結果，其估算的分佈相當地精確。

在作完上述的分析之後，我們只要在編碼端作完運動補償之後，就可以依照所得到的 residual 平均值大小，對於最後編碼完成的 bit-stream 大小作約略的估計，這樣就可以使得最適合的參數組合在更早的階段被決定，其示意圖如下：

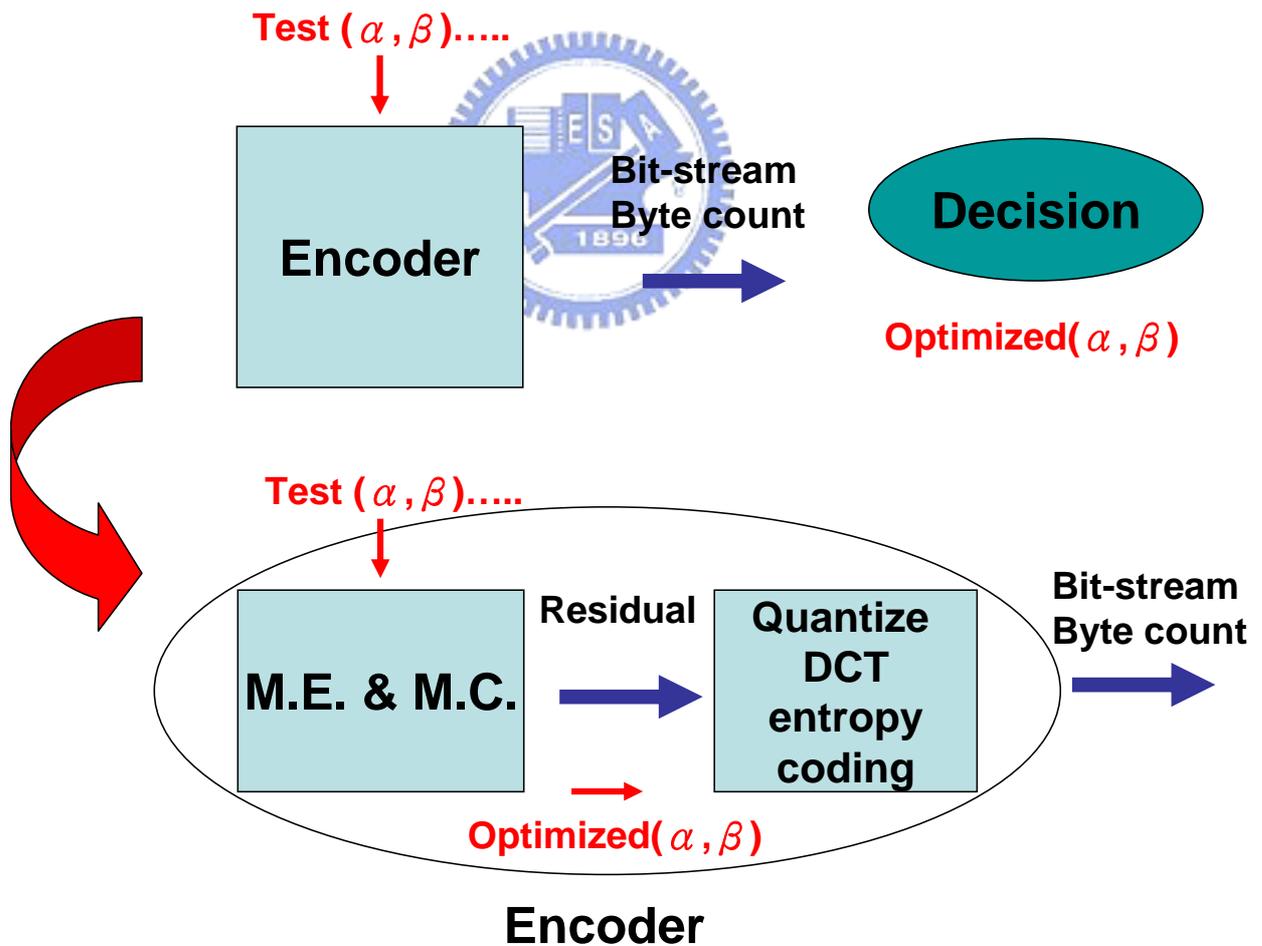


圖 4.20 編碼端參數決定示意圖

第五章 提出的方法

根據第四章對於各項特性的分析，所得到的四個規則，我們將依序建立多種不同的參數組合決定機制，在本章中我們將對於設計的精神想法，和實作的方式作一個介紹和討論。其中，第一節將在固定的頻寬下，提出一個簡單的方法。第二節將在固定的頻寬下，結合之前的四個規則，提出一個完整的參數決定方法。接下來，在第三節中，將在一段可能頻寬下，延伸第二節的方法，提出一個新的作法。第四節中，則是將第三節的判斷機制作一個修改，成爲另一個新的方法。

5.1 方法一

首先，我們先將問題簡單化，考慮當頻寬固定時的參數挑選。因爲從先前的分析中，我們可以了解到錯誤漂移的嚴重性，所以我們決定先不考量錯誤漂移的情形。因爲頻寬是固定的，我們可以經由規則一，預測出最後編碼出來的 bit-stream 情形。在有這兩項資訊的情況下，我們可以選擇適當的 β 值，只讓可以完全被解碼端接收的加強層資訊，被運動補償採用。也就是說，如果頻寬只夠傳輸某一層的部分資料，就算可以傳送 95%，我們爲了避免錯誤的發生，還是不採用。

因爲沒有錯誤的情形會發生在本方法所挑選的參數組合中，所以另外一個參數值 α 就可以盡量地提高，來提升整體的編碼效率。目前，我們把 α 設定在 0.96875。整個設計的想法如下圖所示，實驗的結果將列於下一章中：

Example:

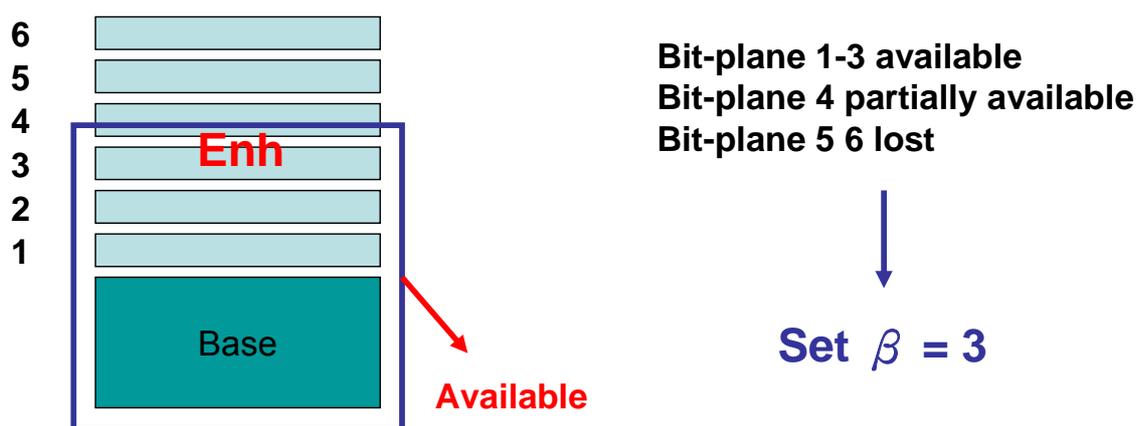


圖 5.1 方法一示意圖

5.2 方法二

很明顯地，在方法一中完全沒有考慮 MSE gain 和 lost 的問題，只是採用最簡單的二分法來決定 β 值，這樣一定會低估了 β 值的挑選，犧牲了太多的編碼效率來確保錯誤漂移不會發生。有鑑於此，我們提出了方法二，同樣是在固定頻寬的條件下，進行參數的挑選。我們另外引進了第四章中所提出的規則二、三、四，來估計某一層的資料量的採用對於整串畫面影像的幫助和損害，也就是 MSE gain 和 MSE lost 的大小。如果幫助大於損害，整體來說 MSE 值會下降，那麼我們就採用這一層的資料量來作為運動補償的參考，也就是把 β 值提升一，以期待整體的編碼效能會提升。示意圖如下：

Example:

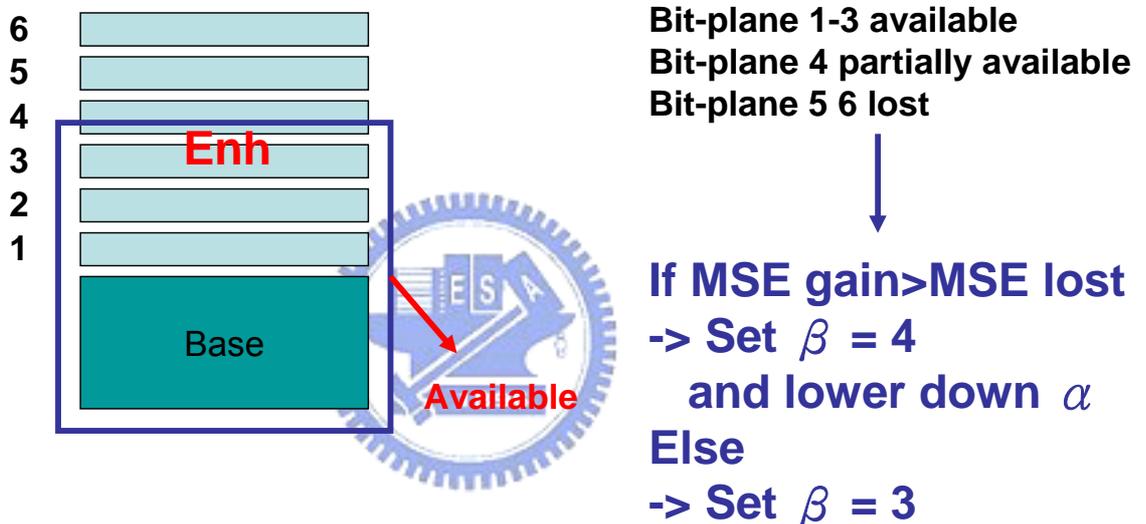


圖 5.2 方法二示意圖

另外，如果說第 k 層的資料被我們評估為不採用之後，那麼第 $k+1$ 層以後的資料就無需再行評估，一定不予採用。因為不可能只採用較高層的加強層資料來作為運動補償的參考。

在另外一個參數 α 的考量上，如果最後 β 的選取是沒有錯誤處罰的問題時，也就是只取用了低於可用頻寬的加強層資料，那麼我們還是把 α 值設定在非常高的位置 0.96875。但是如果當 β 選得過高，也就是說採用了部份無法被解碼端接收的加強層資訊，那麼我們就需要對於接下來的 α 值設定作調整。我們依照遺失的嚴重程度不同，分成兩個情形，當比較輕微時，我們對於接下來的五張畫面之 α 值降低一個 step，如果比較嚴重時，我們對於接下來的五張畫面之 α 值降低兩個 step。這樣對於多張畫面 α 值作調整，相較於只調整一張的 α 值，可以更有效的對於錯誤漂移的情形作控制。本方法的實驗結果，將於下一章中作說明。

5.3 方法三

在考量完單一頻寬的情形後，因為實際的網路情形是會變動的，也就是說我們必須要考量一整段可能的頻寬情形來作編碼，我們取用了和先前研究中一樣的想法，也就是在給定的一段頻寬中，取出數個檢查點，假設每個檢查點的重要性是一致的，在最大化所有檢查點的效果平均值的前提下，來挑選最適合的參數組合。

和方法二的作法非常類似，同樣是採用了規則一到四，對於每一個加強層進行評估。如果對所有的檢查點來說，採用了這一層加強層的資料來作為運動補償的參考可以降低平均的 MSE 值，那麼我們就採用，並把 β 值加一。反之，則不採用，而 β 值不變，而且接下來就不需要考慮更高層的加強層。方法三示意圖如下：

Example:

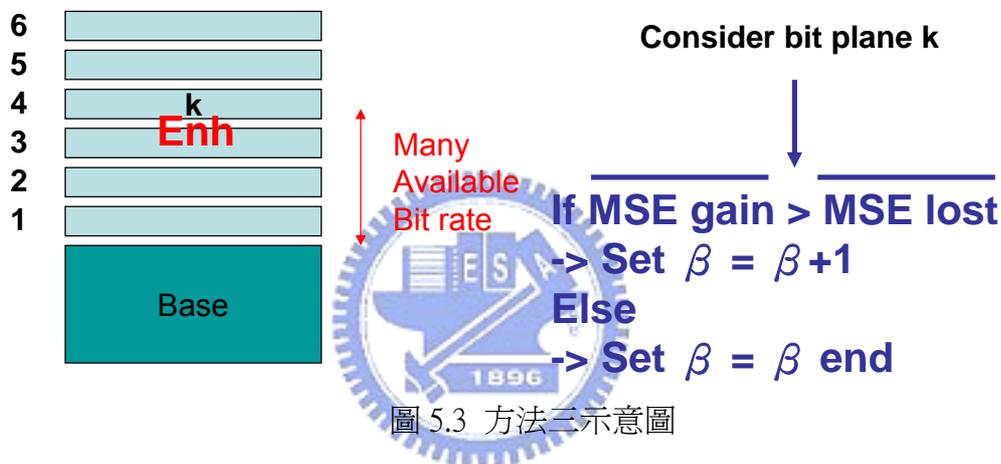


圖 5.3 方法三示意圖

在另一個參數 α 的選取上，和方法二類似。當有超過 1/4 的檢查點採用最後選取的參數值會發生錯誤的情形時，我們把接下來五張畫面的 α 值降低一個 step。如果更嚴重，在超過 1/2 的檢查點會發生錯誤的情形下，我們則把接下來五張畫面的 α 值降低兩個 step。經過這樣的調整，可以有效地控制錯誤漂移的情形。最後得到的結果比較，將於下一章中進行討論。

5.4 方法四

有鑑於在方法三中，在頻寬比較低的檢查點，當採用比較高的 β 值時，會對於 MSE 帶來很大的提升，間接地使得 β 值很難上升。當可能頻寬區間比較小時，這種影響情形較為輕微，但是當考量的可能頻寬區間較廣的時候，也就是說低頻寬檢查點和高頻寬檢查點相差很大的時候，那麼就很容易因為為了顧及少數的頻寬點的情形，而犧牲了大部份的高頻寬檢查點的編碼效能。

為了改善此一問題，我們提出了方法四。MSE gain 和 lost 等等的評估，和方法三一樣，但是在決定這一層的加強層資訊是否採用時，決策的機制不一樣，改成考量帶來好處和壞

處的檢查點個數，如果超過半數的檢查點在採用了這一層的加強層資料量之後，可以對 MSE 的下降帶來幫助，那麼我們就決定多採用這一層的加強層作為運動補償的參考，把 β 值提升一。相反的，如果過半數的檢查點會因為採用了這層多出來的加強層資訊而產生 MSE 上升的情形，那麼我們就不採用， β 值維持不變。如下所示：

Example:

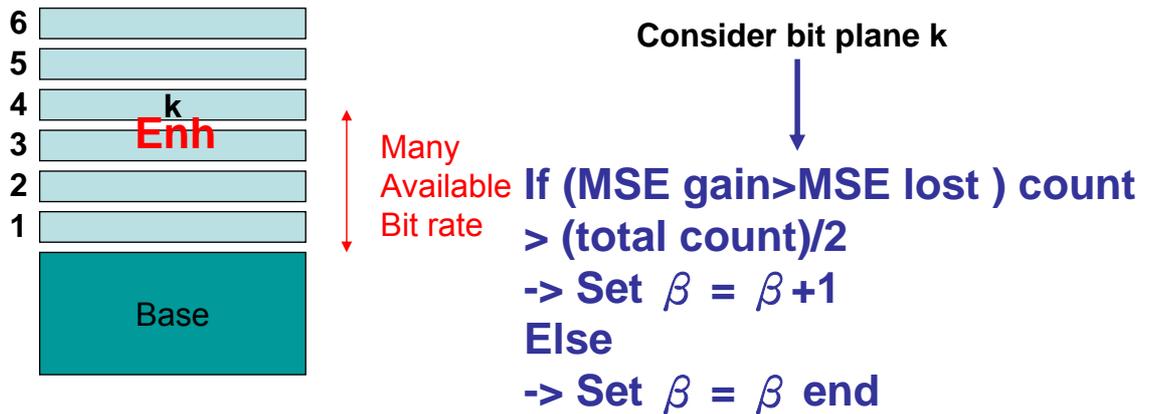


圖 5.4 方法四示意圖

而在另一個參數 α 的挑選上，維持和方法三中一樣的挑選方式。最後的實驗結果將於下一章中被提出並比較討論。

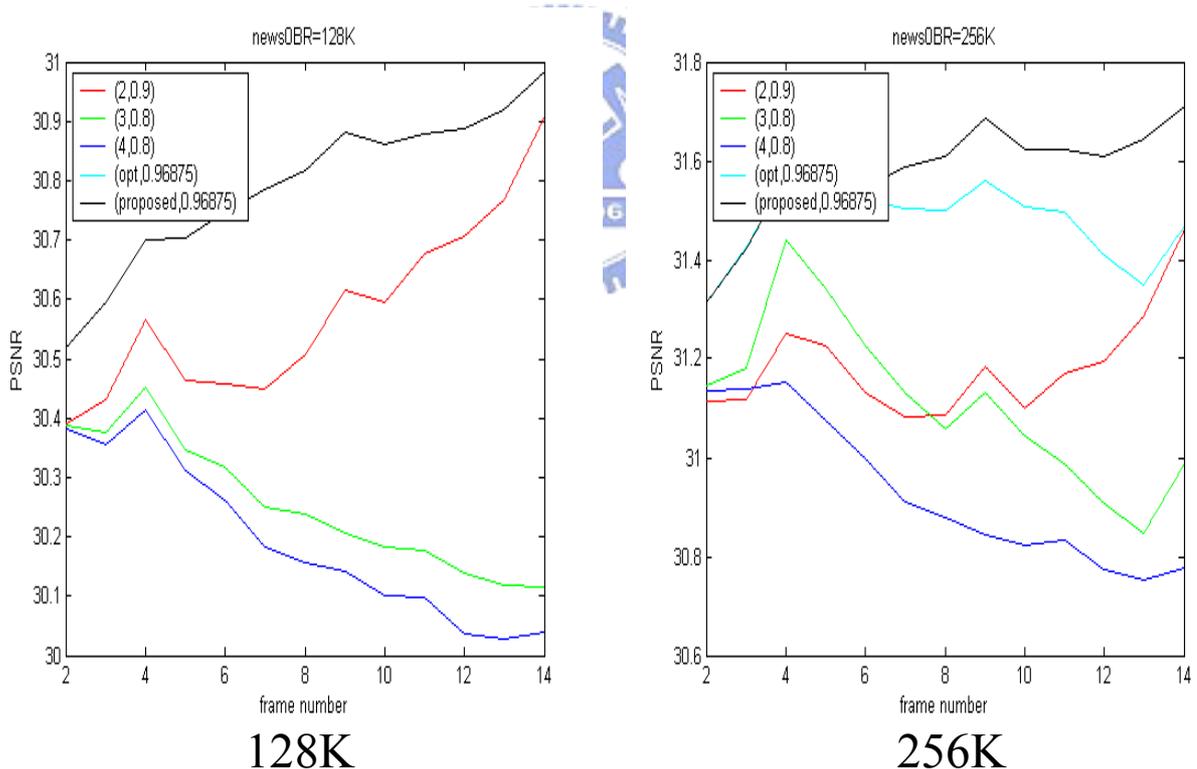


第六章 實驗結果

在本章中，將比較上章中各項方法的結果進行比較，並且討論效能提升或者是下降的原因，測試的影像主要為 News、Foreman、Stefan 分別代表了慢速度，中速度和快速度的影像，頻寬考量是從 256K 到 2560K，畫面率設為 30。

6.1 方法一實驗結果

如第五章中所介紹的，本方法是用最簡單的想法來實現一個不會受到錯誤漂移影響的參數挑選方式。在固定頻寬下，因為我們只採用了可以被解碼端收到的資訊量來作為運動補償的參考，雖然沒有辦法達到比較好的編碼效率，但是可以免除掉錯誤帶來的處罰，在比較低頻寬的時候，可以帶來比較好的效果，接近於之前最佳參數挑選的結果。我們一共在 256K 到 2560K 之間，分別測試了 256K、384K、512K、768K、1024K、1536K、2048K 和 2560K 等八個頻寬，部份的結果如下圖所示：



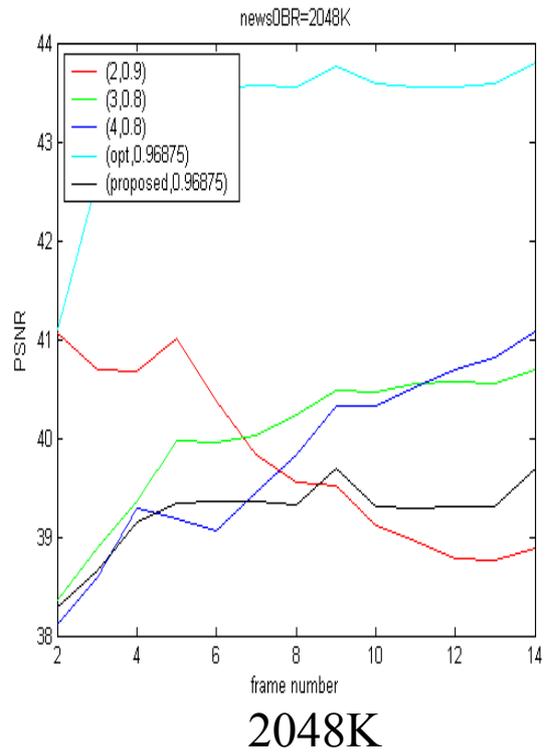
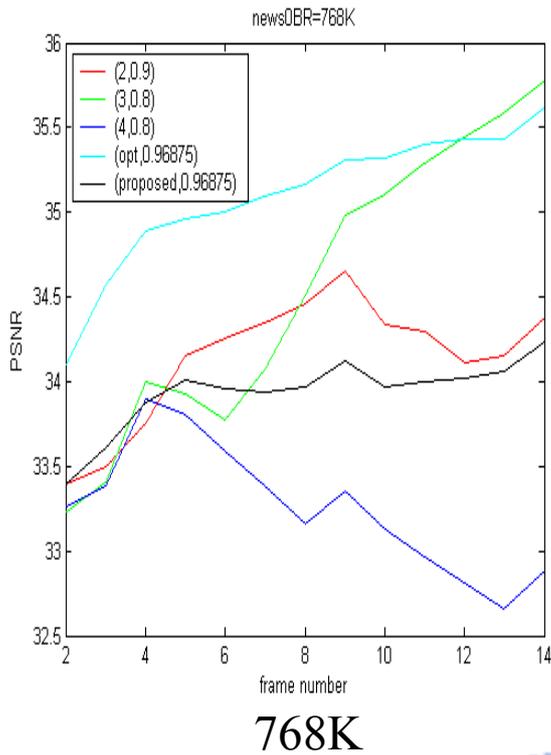


圖 6.1 方法一結果圖

6.2 方法二實驗結果

在方法二中，我們開始使用我們發現的規則，對於解碼端影像的品質進行評估，希望在某些適合的情形下，採用部份解碼端不會被收到的資訊來作為運動補償之參考，希望更進一步的提升編碼品質，同樣測試了 256K、384K、512K、768K、1024K、1536K、2048K 和 2560K 等八個頻寬，其結果如下圖所示：

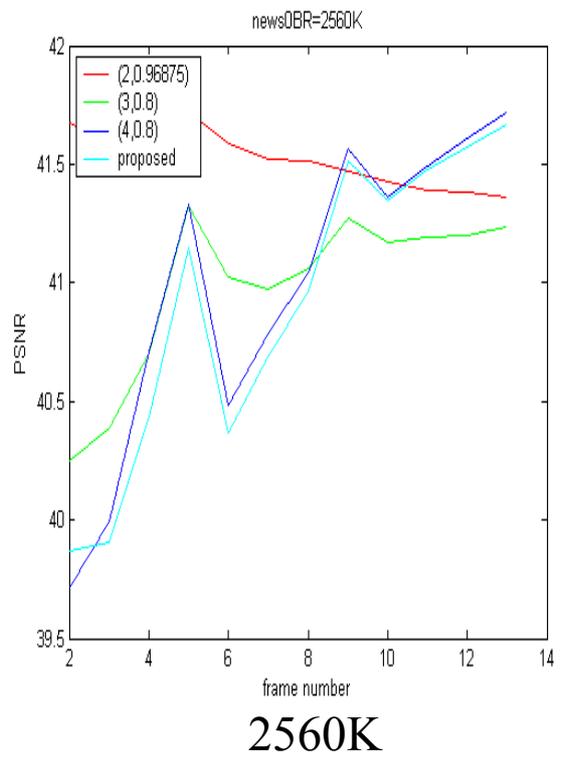
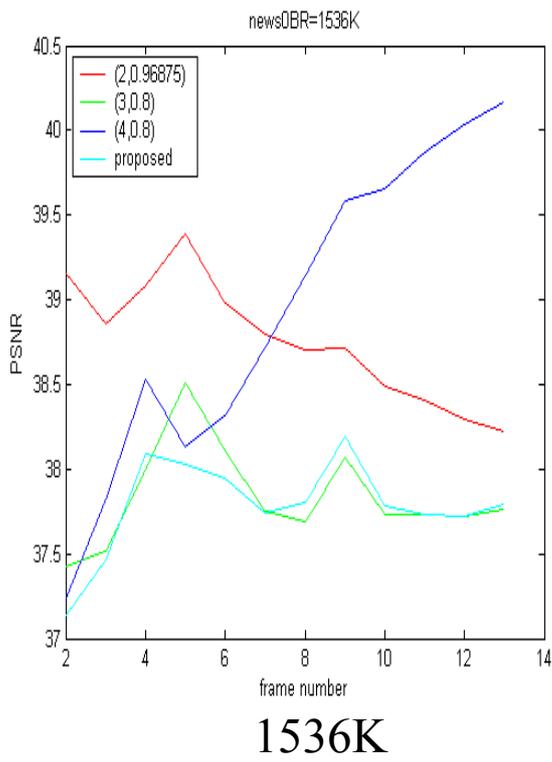
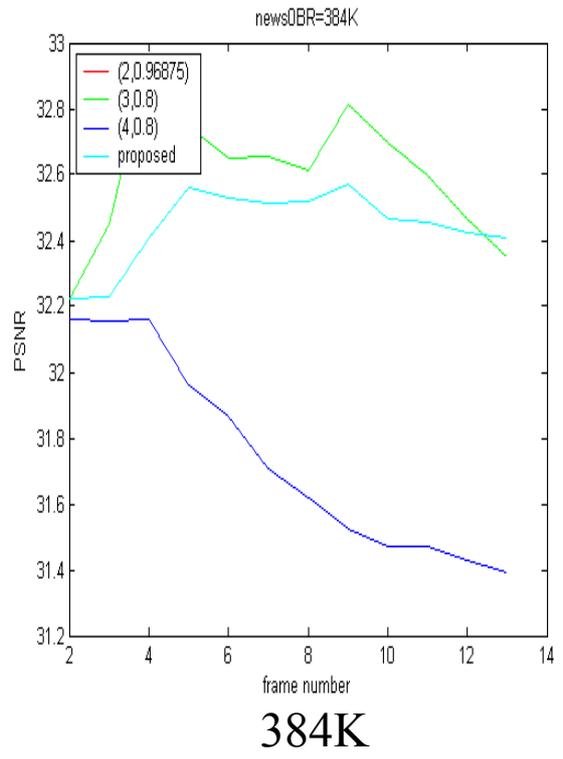
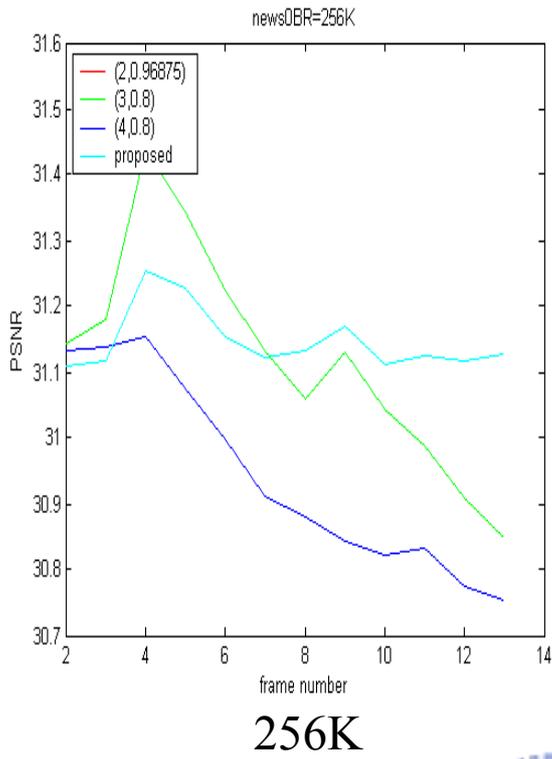


圖 6.2 方法二結果圖

6.3 方法三實驗結果

在方法三中，延續方法二的估測方式，只是改為採用一整段的頻寬來考量，經過實驗結果發現，如果考量的頻寬較長的時候，因為在低頻寬時錯誤對於 MSE 的影響非常的大，會主宰參數的決定，所以最後挑選出來的結果就幾乎都是選擇 β 為 2，這並不是我們希望的結果。但是如果考量的頻寬範圍較小，避開一些非常低的頻寬，就可以帶來比較好的效果，我們取用了 768K、896K、1024K、1280K、1400K、1536K、1750K 和 2048K 等八個頻寬作為檢查點，得到如下的效果：

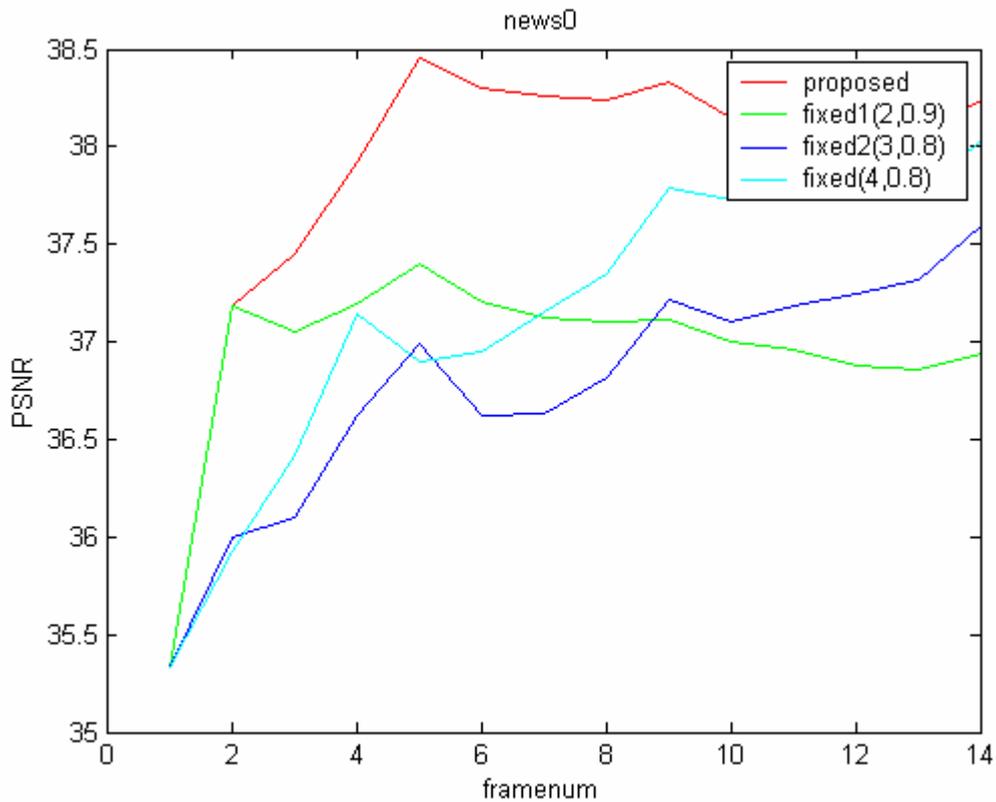


圖 6.3 方法三結果圖(News)

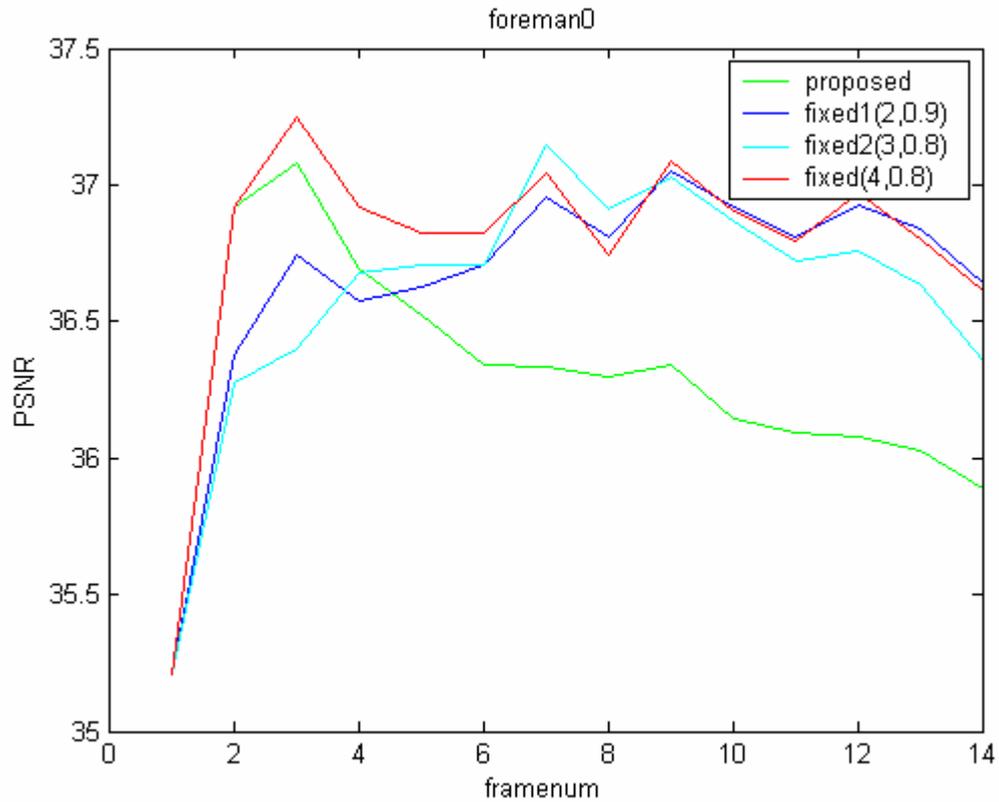


圖 6.4 方法三結果圖(Foreman)

6.4 方法四實驗結果

最後爲了改善方法三在比較大的可能頻寬，含蓋了低頻寬時，挑選參數組合結果不佳的問題，我們改採了以改善與否的數量取代了平均改善量，作爲衡量的標準，一樣是考量 256K、384K、512K、768K、1024K、1536K、2048K 和 2560K 等八個頻寬，得到的結果如下：

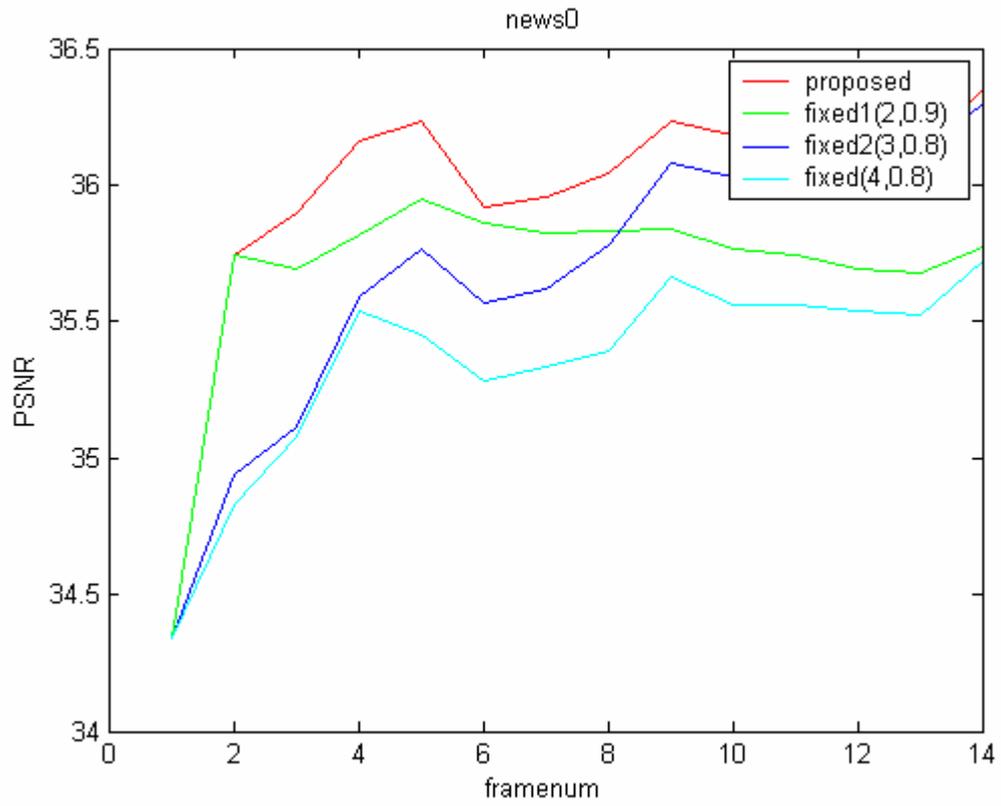


圖 6.5 方法四結果圖(News)

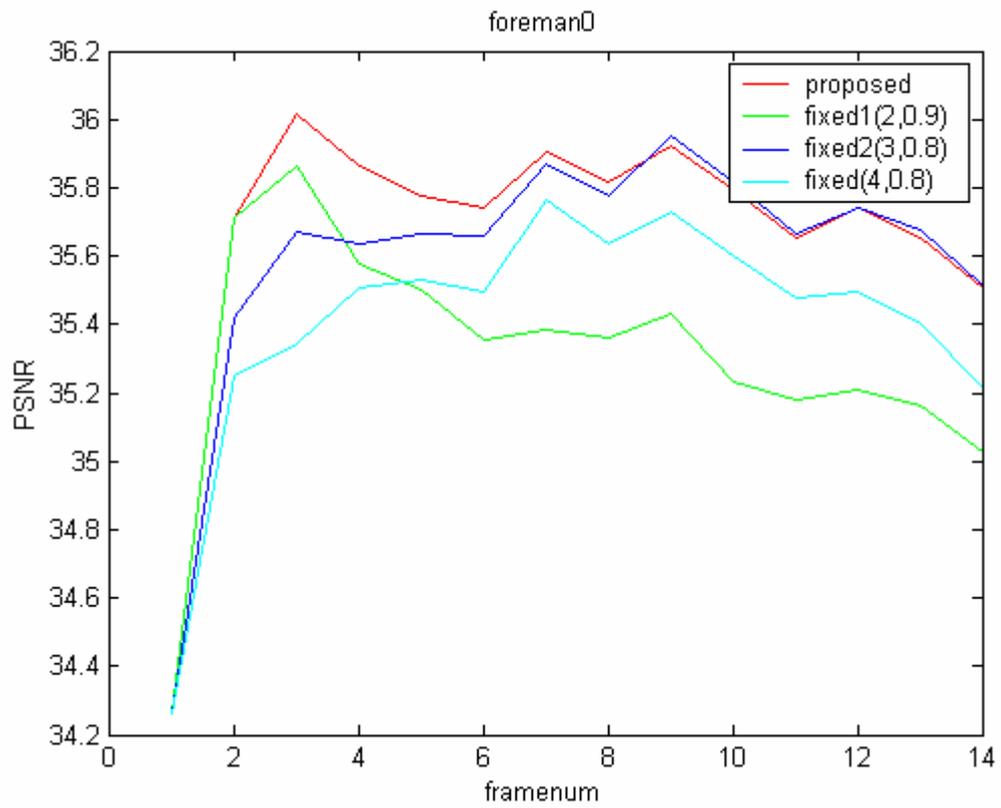


圖 6.6 方法四結果圖(Foreman)

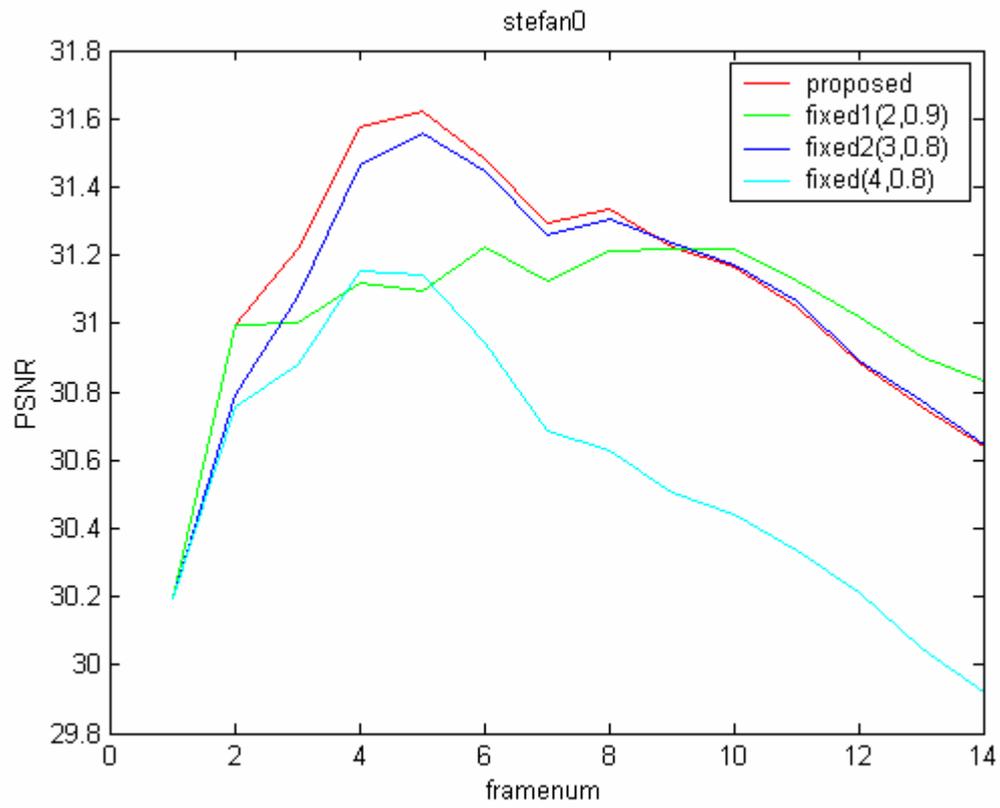


圖 6.7 方法四結果圖(Stefan)



第七章 結論

在本論文中首先分析原本以 PSNR 為標準的 RFGS 最佳化參數挑選方法，為了達到平衡的考慮低、高頻寬情形，並且改善過度低估錯誤處罰的缺點，提出了以 MSE 為標準的 RFGS 最佳化參數挑選方法，相較於原本的方法，提升了一定程度的整體編碼效果。有鑑於上述的兩種方法，需要經過多次的編碼，通道，解碼才能得到最佳參數組合，雖經過加速，但是速度仍然非常的慢，在實作上幾乎不可行。我們接下來觀察了編碼，解碼端各過程的結果，經過統計分析，發現其規律，建立各個相對應的模型;最後，藉由上述的結果，我們提出了一些不同的快速演算法，可以在編碼端經過簡單的一些運動補償測試，就可以估算出一組最適合的參數組合。

我們提出以 FRAME 為基礎的完整方法，相對於固定參數的編碼方式，在增加非常少的編碼時間之下，因為對於有不同特質的畫面，採用不同的參數組合，可以一定程度的提升了編碼效果。



參考文獻

- [1] *Coding of Audio-Visual Objects, Part-2 Visual, Amendment 4: Streaming Video Profile*, ISO/IEC 14 496-2/FPDAM4, July 2000.
- [2] Mihaela van der Schaar, Hayder Radha, "A hybrid temporal-SNR Granular Scalability for Internet Video," *IEEE Trans. On Circuits Syst. Video Technol.*, vol. 11, No. 3, March 2001
- [3] Feng Wu, Shipeng Li, Ya-Win Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. On Circuits Syst. Video Technol.*, vol. 11, No. 3, March 2001
- [4] Weiping Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. On Circuits Syst. Video Technol.*, vol. 13, No. 3, March 2001
- [5] Jyh-jyh Huang, "Visual Improvement on MPEG-4 Video Coding with Frequency Information Selection", Department of Electronics Engineering, Institute of Electronics National Chiao Tung University Res. 2004
- [6] Mihaela van der Schaar, Hayder Radha, "Motion-compensation fine-granular-scalability (MC-FGS) for wireless multimedia" *Proceeding Multimedia Signal Processing, IEEE 4th Workshop on*, Oct. 2001
- [7] Hsiang-Chun Huang, Chung-Neng Wang, and Tihao Chiang, "A Robust Fine Granularity Scalability Using Trellis-Based Predictive Leak," *IEEE Trans. On Circuits Syst. Video Technol.*, vol. 12, No. 6, June 2002
- [8] S.R. Chen, C.P. Chang, and C.W. Lin, "MPEG-4 FGS coding performance improvement using inter-layer prediction," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 2004, Montreal, Canada.
- [9] Y. Liu, Z. Li, Paul Salama, and E. J. Delp, "A discussion of leaky prediction based scalable coding," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, pp. 565-568, vol. 2, July 6-9, 2003, Baltimore, MD.
- [10] L.C. Kuo, T. Chiang, S.J. Wang, "An Adaptive Macro-Block Based Scheme for Temporal Prediction Control in Fine Granularity Scalability Coding," in *Conf, on WCE*, Nov 2004, Hsinchu, Taiwan