

國立交通大學

資訊科學與工程研究所

碩士論文

基於最佳停止理論所設計之可調視訊多層編碼
控制的快速編碼模式決策演算法

A Fast Mode Decision Algorithm based on Optimal Stopping

Theory for SVC Multi-Layer Encoder Control

研究生：黃嘉彥

指導教授：彭文孝 教授

中華民國 一 百 年 十 月

基於最佳停止理論所設計之可調視訊多層編碼控制的

快速編碼模式決策演算法

研究生：黃嘉彥

指導教授：彭文孝

國立交通大學資訊科學與工程研究所 碩士班

摘要

基於可調視訊編碼(SVC)之多層編碼控制(Multi-layer Encoder Control, MLEC)的架構，本論文提出一個快速編碼模式決策的演算法。此提出的演算法主要在於改善 Zhao's *et al.* 應用最佳停止理論(Optimal Stopping Theory, OST)所提出的快速編碼模式決策演算法中所存在的問題。其演算法的問題是將所有編碼模式(mode)的效益都視為相同，只考慮編碼模式序列中下一個相對更好的編碼模式的位置和目前在序列中停止的位置相距多遠，但此距離並不能給予任何保證。不能保證這個相對更好的編碼模式可以得到多少 RD-cost 的下降，而且也不能保證需要花費多少運算複雜度。因此，本論文將以上兩點加以考慮，設計出一套更有效率的快速編碼模式決策演算法。

本論文將分析編碼模式配對序列中的不同停止點，若往下找到一組相對更好的編碼模式配對時能得到多少 RD-cost 下降量，以及分析不同的編碼模式配對在編碼時各需要多少運算複雜度。再將以上兩項資訊建為表格，提供在計算最佳停止點時所使用。除此之外，本論文還提出一個精進編碼模式決策的方法，利用統計與分析 RD 資訊找出較差的編碼模式配對，進一步執行精進程序。實驗結果顯示，本論文所提出的快速編碼模式決策演算法和徹底式搜尋(exhaustive search)所需編碼時間相比，節省了編碼時間逾 80%~85% 左右且有相當的 RD 效能。此外，本論文所提出的演算法和其他兩種不同的演算法相比，亦也有更好的穩定性，以及編碼品質。

A Fast Mode Decision Algorithm based on Optimal Stopping Theory for SVC Multi-Layer Encoder Control

Student: Jia-Yan Huang

Advisor: Wen-Hsiao Peng

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract

This thesis proposed a fast mode decision algorithm for scalable video coding (SVC) multi-layer encoder control (MLEC). The proposed algorithm attempts to improve the problem of the fast algorithm proposed by Zhao's et al. based on the optimal stopping theory (OST), in which the problem is the benefit obtained of the different mode is regarded as the same and only the distance considered between the stopping point and the position of the next relative best mode in the mode sequence. Unfortunately, this distance cannot guarantee the amount of RD-cost decrease in the next relative best mode; additionally, the amount of required computational complexity to the next relative best mode cannot be anticipated either. Thus, this thesis will take aforementioned points into consideration in order to create a more efficient fast mode decision algorithm.

This thesis analyzed the stopping point in the mode pair sequence, which explore the amount of RD-cost decrease by finding the next relative best mode pair, and analyze the different amount of computational complexity between the different mode pairs. A table containing the above two points will also be provided for calculation of the optimal stopping point. Additionally, this thesis will proposed a more refined encoding method by utilizing statistics and data analysis to identify inferior encoding mode pairs and then refining them. Experimental result demonstrates that the proposed fast mode decision algorithm compares to the method of exhaustive search can achieve time saving about 80% to 85% with negligible quality degradation. Furthermore, the proposed method is superior to those of the other two types because of its stability and encoding quality.

致謝

在兩年的碩士生涯內，首先我要感謝我的指導教授—彭文孝 博士。彭老師在兩年內不論在學問研究上以及做人處事上都給予我精闢的指導。彭老師教導我做研究應該要有的態度以及遇到問題如何解決，並且要求我們做於研究生應該有的態度。在老師細心以及耐心的指導下，讓我能最有效的時間內完成碩士論文並且在兩年內受益無窮。在此再一次感謝我的彭老師，並致上無限的敬意。

其次，這篇論文可以完成，我也要感謝林哲永學長和蔡閔旭學長在我的研究上擔任了啟蒙的指導，不僅在 H.264 和 SVC 的專業領域上，不辭辛勞的與我討論，更在一開始的研究實作上給予許多珍貴的意見，並且能適時從旁給予建議修正我已偏差的研究方向，使我在這兩年的碩士生涯，不再舉步維艱。謹此致上由衷的謝意。

非常榮幸進入 MAPL 實驗室，能夠有熱心與親切的實驗室成員們的切磋與討論，是我在碩士時期最充實的時光。我要感謝其餘的學長姐們—陳漪紋 博士、陳俊吉 博士、與詹家欣 博士、王澤瑋、楊復堯、與吳思賢，一步步帶領我進入這個專業的領域；感謝我的好同學們吳崇豪、李宗霖、陳孟傑和曾于真，不論是研究上或是生活上，他們總是給予我最直接的協助以及苦樂分享；感謝我的學弟吳牧軒、陳彥宇，提供我實驗上適時的幫助。

最後，我要感謝我的父母—黃富村 先生、石玉琴 女士的栽培，在取得碩士學位的路途上，給予精神上和物質上的支持和溫暖，讓我能夠專心的在研究領域上打拼。感謝我的哥哥—黃嘉維，以及我的姐姐—黃美嘉，也能適時的提供我意見，並且不斷的鼓勵我繼續向前。感謝你們一路上的陪伴打氣與支持，在此僅將這篇論文獻給各位，謝謝你們！

目錄

摘要.....	i
Abstract.....	ii
致謝.....	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	vii
第一章 緒論	1
1.1 研究背景.....	1
1.1.1 可調視訊編碼之 CGS 架構.....	1
1.1.2 最佳停止理論—Duration Problem.....	2
1.1.3 Duration Problem in 編碼模式決策.....	2
1.2 問題闡述與提出解決辦法.....	4
1.3 研究貢獻.....	5
1.4 論文編排.....	5
第二章 相關研究	6
2.1 可調視訊編碼.....	6
2.1.1 基本架構.....	6
2.1.2 層與層之間預測.....	7
2.2 可調視訊編碼之編碼控制.....	9
2.2.1 由下而上的編碼控制.....	9
2.2.2 多層編碼控制.....	10
2.2.3 編碼控制應用於編碼模式決策.....	11
第三章 快速編碼模式決策演算法	13
3.1 編碼模式決策的兩項因素.....	14
3.1.1 考慮 RD-cost 下降量.....	15
3.1.2 考慮編碼模式配對的運算複雜度.....	17
3.1.3 同時考慮 RD 和運算複雜度，並找出停止點.....	20
3.2 編碼模式配對的序列產生.....	24
3.3 編碼模式決策的精進方法.....	26
3.3.1 判定 RD 效能.....	26
3.3.2 精進的方法.....	28
3.3.3 精進程序前後差異比較.....	30
3.4 快速編碼模式決策演算法的流程.....	32
第四章 實驗結果與討論	33
4.1 實驗環境.....	33

4.2 實驗比較與討論.....	34
4.2.1 實驗一.....	34
4.2.2 實驗二.....	38
4.2.3 實驗三.....	41
第五章 結論與未來展望	42
參考文獻.....	43



圖目錄

圖 1	可調視訊編碼的 CGS 架構.....	1
圖 2	編碼模式序列.....	3
圖 3	可調視訊編碼所提供的三種不同調節性，由左至右分別為(1)空間可調性(2)時間可調性(3)畫質可調性.....	6
圖 4	層與層之間的移動向量預測示意圖.....	7
圖 5	層與層之間的殘值預測示意圖.....	8
圖 6	與層之間的畫面內預測.....	8
圖 7	(a)傳統由下而上編碼控制應用於編碼模式決策示意圖 (b) 多層編碼控制應用於編碼模式決策示意圖.....	12
圖 8	編碼模式配對示意圖.....	12
圖 9	序列中編碼模式配對的運算時間複雜度示意圖.....	17
圖 10	reward function $-y_k$ 趨勢示意圖 BLQP=22, ELQP=18, W = 0.1	20
圖 11	基礎層在不同規範值的位元率上升圖.....	21
圖 12	增進層在不同規範值的位元率上升圖.....	21
圖 13	基礎層在不同規範值的 PSNR 值.....	22
圖 14	增進層在不同規範值的 PSNR 值.....	22
圖 15	在不同規範值的編碼時間節省.....	23
圖 16	利用哪些 MB 產生編碼模式配對序列示意圖.....	25
圖 17	演算法流程步驟圖.....	32

表目錄

表 1	測試的編碼模式種類表.....	13
表 2	RD-cost 下降量表	16
表 3	各種編碼模式配對的運算時間表.....	18
表 4	不同設定下 y_k 的規範值表	21
表 5	目前編碼的 MB 和周圍 MBs 做相同編碼模式配對的機率表	24
表 6	目前編碼 MB pair 和前一張畫面相對位置 MB pair 的 RD ratio 差值.....	27
表 7	精進方法的編碼模式配對順序表—固定基礎層.....	29
表 8	精進方法的編碼模式配對順序表—固定增進層.....	30
表 9	序列中編碼模式配對種類的比較.....	30
表 10	實驗環境設定.....	33
表 11	BLQP = 22 ; ELQP = 18 不同演算法的比較.....	34
表 12	BLQP = 26 ; ELQP = 22 不同演算法的比較.....	36
表 13	BLQP = 30 ; ELQP = 26 不同演算法的比較.....	37
表 14	BLQP = 22 ; ELQP = 18 Zhao's 演算法中加入所提出的精進方法	38
表 15	BLQP = 26 ; ELQP = 22 Zhao's 演算法中加入所提出的精進方法	39
表 16	BLQP = 30 ; ELQP = 26 Zhao's 演算法中加入所提出的精進方法	40
表 17	其他不同的視訊序列 RD 效能.....	41

第一章 緒論

1.1 研究背景

由於網路傳輸以及多變容量不同設備的蓬勃發展，視訊內容在各種應用上的可調性已經是重要且非常重要的問題。基於這個理由，Joint Video Team(JVT)以H.264/AVC為基礎，制訂了可調視訊編碼(Scalable Video Coding, SVC)[1]。可調視訊編碼在單一位元串流中提供了不同的調節性，其中包括 1)空間可調性(spatial scalable)、2)時間可調性(temporal scalable)、3)品質可調性(quality scalable)。可調視訊編碼是由一個基礎層(base layer, BL)和多個增進層(enhancement layer, EL)所組成，且相較於低層的增進層是相較於高層增進層的基礎層，然而增進層利用層與層之間預測的方式(inter-layer prediction)，盡可能的使用基礎層的資訊來達到減少壓縮時所需要的位元率。

1.1.1 可調視訊編碼之 CGS 架構

粗略可調編碼是(Coarse-grain Quality Scalable Coding, CGS)可調視訊編碼的品質可調性中的一種架構，此架構最基本的設計為基礎層和增進層擁有相同的畫面解析度(resolution)，兩層之間利用不同量化參數(QP)的設定來達到不同層之間的調節作用，以圖 1 為例。圖 1 是可調視訊編碼中的 CGS 架構且畫面序列之間是 IPPP 架構，從圖中可以看到此架構的上下兩層是由相同畫面解析度所形成，

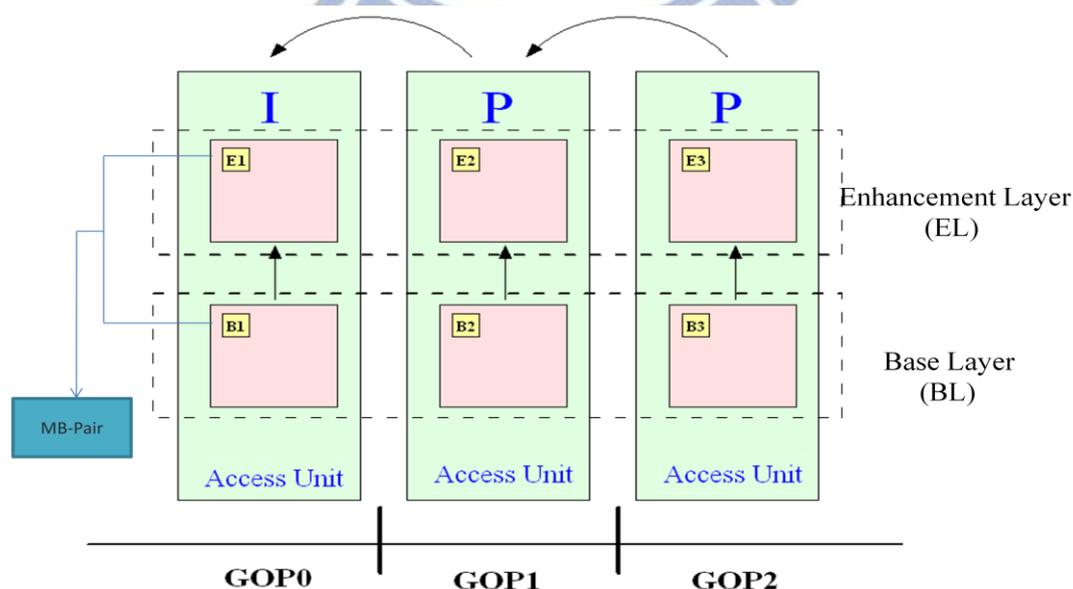


圖 1 可調視訊編碼的 CGS 架構

且基礎層做為增進層的預測。圖中兩層同一張畫面之間相同位置的 MB 稱為一組 MB pair，例如(B1,E1)就是一組 MB pair。這組 MB pair 在可調視訊編碼中扮演非常重要的腳色，MB pair 中的兩個 MB 之間的編碼資訊有某種程度的相關性，由於增進層利用基礎層做預測，因此基礎層在編碼時參數的設定亦會牽動增進層的編碼參數設定。以編碼模式決策(mode decision)為例，兩層之間的編碼模式亦會有很大的關連性，若基礎層所選擇的編碼模式為 Inter16x8，則增進層的編碼模式也有較大的機會是相同的，且編碼模式為 Inter8x16 的可能性就非常的小，因為此兩種編碼模式的分割方式差異較大。

1.1.2 最佳停止理論—Duration Problem

最佳停止理論(Optimal Stopping Theory, OST)[2]中存在一個最佳停止問題，此問題主要是在說一位決策者如何逐一訪談一連串的物件中，每訪談完一個物件後決定是否要繼續訪談下一個物件。然而這項理論主要是由兩項物件所形成，1) 一連串的隨機變數、2)一個用來計算分數的函式。以面試官的例子說明，若有一連串的人員逐一等著面試，這裡要被面試的人員可視為隨機變數，因為每個人擁有不同的工作經歷或學歷。面試官想在人群裡挑出一位優秀的人員加入，但又不想將所有人員都面試完。此時面試官利用一套計算分數的函式(稱為 reward function)，去計算看說面試到哪位人員停止，能得到最高的效益。然而在最佳停止理論中，成功找到最佳停止點的條件只有真的在序列中找到最好的物件，否則都算是失敗的停止點。

然而，並不是所有的問題都是必須找到最佳的物件。可以依照問題的目的，找到相對較好的物件即可，這裡以快速編碼模式演算法為例，此問題的目的是為了節省編碼所需的時間，避免使用完整搜尋的編碼模式決策方法去測試所有可能的編碼模式並找出一個最佳的，且在壓縮位元率和畫面品質上不能和完整搜尋法有太大的差異。簡單的說，快速編碼模式決策演算法是去找出相對較好的編碼模式即可。因此，基於這些只需要找到相對較好物件的問題，最佳停止理論將這類型的問題稱為 Duration Problem [3]。在最佳停止理論中的 Duration Problem，其計算分數的函式是用來計算序列中，下一個會遇到相對更好的物件需要花費多少代價。因此，在此問題的最佳停止點就是該停止點遇到下一個更好的物件需要花費最大代價的點即是。所以 Duration Problem 成功找到停止點的定義跟最佳停止問題不太一樣，這個問題找到的停止點只要能滿足你問題的目的即可。

1.1.3 Duration Problem in 編碼模式決策

在此小節，介紹如何將 Duration Problem 應用於編碼模式決策上，此應用是由 Zhao's *et al.*所提出[4]。Zhao's *et al.*[4]利用 Duration Problem 在可調視訊編碼

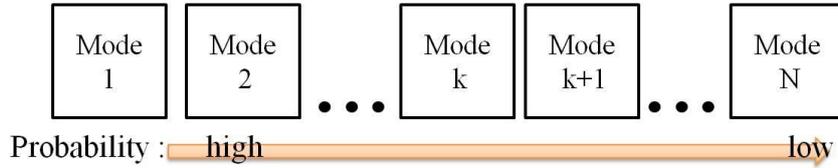


圖 2 編碼模式序列

的編碼模式決策上設計出一套編碼模式提前結束的快速演算法，此方法如圖 2 所示。Zhao's *et al.* [4] 利用目前要編碼 MB 周圍的資訊產生一組編碼模式序列，其排序方式是將目前要編碼的 MB 最有可能的編碼模式，依機率由大到小排序。此排序的目的，是為了讓 reward function 擁有 unimodal function 的特性，也就是要確保此 reward function 只擁有一個極大值。

在上一小節中提過，Duration Problem 是去找出下一個相對更好的物件時需要花費多少代價。同樣的，在編碼模式決策時也是如此，在序列中的每一個編碼模式，都可以利用 reward function 去計算下一個遇到更好的編碼模式需要花費多少代價。在 Zhao's *et al.* [4] 所提出的演算法中，此代價是定義為距離目前停止點多遠的距離，也就是去計算遇到下一個相對更好的編碼模式需要再做多少個編碼模式。若以方程式來說明

$$E(T_k) = \sum_{j=k+1}^N [j \cdot P(T_k = j)] + (N + 1) \cdot P(T_k = N + 1) \quad (1)$$

and

$$y_k = \frac{E(T_k) - k}{N} \quad (2)$$

其中 T_k 代表目前停在第 k 個編碼模式且下一個遇到更好的編碼模式是在序列中的第幾個，特別是 $T_k = N + 1$ 代表在第 k 個編碼模式後不會出現相對應更好的編碼模式。而 $E(T_k)$ 則是去計算 T_k 的期望值， y_k 就是所謂的 reward function，去計算下一個預期出現更好的編碼模式位置和目前停止位置 k 距離多遠。在 Zhao's *et al.* [4] 所提出的演算法中，在編碼模式序列中每一個位置都可以去計算一個 y_k ，而此演算法停止的位置就是擁有最大 y_k 的編碼模式。

1.2 問題闡述與提出解決辦法

在 1.1.3 小節中提到 Zhao's *et al.*[4] 將 Duration Problem 應用在編碼模式決策上並設計出一套快速編碼模式決策的演算法。但是 Zhao's *et al.*[4]所提出的演算法存在了兩項問題，其問題分別為：

- ◇ 沒有考慮停在不同位置的編碼模式時，下一個遇到更好的編碼模式能獲得多少 RD-cost 下降量。
- ◇ 沒有考慮不同編碼模式的運算複雜度。

因為在 Zhao's *et al.*[4]所提出的演算法中只是去計算每一個不同停止位置與下一個相對更好的編碼模式距離多遠，但這個距離並不能給予任何的保證，不能保證距離越遠就代表能得到越多的 RD-cost 下降量，亦不能保證距離越遠就代表其所需編碼時間就越長。在 Zhao's *et al.*[4]所提出的演算法中，每一個不同位置下一個遇到更好的編碼模式都視為可以得到相同的 RD-cost 下降量，而且不同的編碼模式，其所需的編碼時間也都視為相同的，但這跟現實情況真的相同嗎。以不同編碼模式所需的編碼時間為例，在 Inter16x16 和 Inter8x8 兩種不同的編碼模式，直覺上一定認為 Inter8x8 所需的編碼時間一定比 Inter16x16 所需的多，且實際情形也是如此，因此怎麼能只考慮下一個遇到更好的編碼模式要再做多少個編碼模式呢。

基於上述兩個問題點，本篇論文進一步加以考慮並改善 Zhao's *et al.*[4]所提出的方法。首先，因為每一個編碼模式序列中不同的停止位置，可能會依照在序列中位置的不同而下一個遇到相對更好的編碼模式所得到的 RD-cost 下降量也有所不同的這一點考慮，作者認為必須去計算若目前停止位置為 k ，則下一個遇到相對更好的編碼模式會有多少期望的 RD-cost 下降量，這裡記做為 $E(RD_{T_k})$ 。再來，還要去計算下一個會遇到更好的編碼模式，途中所做過的不同編碼模式總共需要多少期望的編碼時間，這裡記做為 $E(C_{T_k})$ 。然而，這兩個期望值 $E(RD_{T_k})$ 、 $E(C_{T_k})$ 分別可以視為下一個遇到更好的編碼模式能獲得多少好處，以及需要花費多少代價。最後，將這兩個期望值一起考慮為一個比值 $y_k = E(RD_{T_k})/E(C_{T_k})$ ，這個比值稱為獲得好處與需要花費比，作者利用這個比值去找出在編碼模式序列中，停在哪一個位置能得到最佳的效益。至於如何利用這個比值去找出最佳的停止位置，將在之後的章節會詳細的說明。

1.3 研究貢獻

在這次的研究工作中，最主要的貢獻是提出一套新的快速編碼模式決策演算法，其中包含以下幾點：

- ◇ 有效的利用周圍的資訊去產生編碼模式配對序列，以提供目前的 MB 所使用。
- ◇ 有效的改善原先 Zhao's *et al.*[4]所提出演算法中的問題，在此利用統計與分析的方式成功建立不同位置的 RD 下降表，以及提供不同編碼模式配對的編碼運算複雜度參考。
- ◇ 有效的提出一套精進編碼模式的機制，不但能在作者所提出的演算法上實施，在其他的演算法中也能適用。
- ◇ 作者提出的快速演算法能有效的在編碼時間上加速，而且位元率與畫面品質和完整搜尋法相比，並沒有太大的上升與失真。

實驗結果顯示，作者所提出的快速編碼模式決策演算法和完整搜尋法比較，能得到 80% ~85% 的編碼時間節省，並且擁有相似的位元率—失真效能。並且作者所提出的精進編碼模式的方式，亦能有效改善 Zhao's *et al.*[4]所提出演算法的位元率—失真效能。

1.4 論文編排

本論文之後的組織架構為：第二章包含可調視訊編碼的基本架構與編碼控制的介紹。第三章呈現作者所提出的快速編碼模式決策演算法。第四章為實驗結果與討論。最後，第五章為本篇論文的結論以及未來展望。

第二章 相關研究

2.1 可調視訊編碼

2.1.1 基本架構

可調視訊編碼 (Scalable Video Coding, SVC) 是 Joint Video Team (JVT) 以 H.264/AVC 為基礎進而延伸出來的另一套標準，讓視訊編碼能附有調節的功能。可調視訊編碼主要是由一個基礎層(BL)和多個增進層(EL)所組成，且位於較低層的增進層是位於較高層增進層的基礎層。層與層之間的可調性，能達到提供給各種不同的環境和不同的機器設備，且在可調視訊編碼的架構下只需將多層不同品質的視訊編碼成單一位元串流，當環境或機器的限制下，若只需要某層的視訊品質，只要擷取單一位元串流中部份的位元串流即可得到此層所提供的視訊品質。因此，可調視訊編碼的可調節性，可以有效的提供給不同的機器設備或環境，然而在可調視訊編碼中主要支援三種可調性，其中包含：

- ◇ 空間可調性 (Spatial Scalability)
不同層之間提供不同畫面解析度來做調節。
- ◇ 時間可調性 (Temporal Scalability)
不同層之間提供不同每秒畫面張數來做調節 (frame per second)。
- ◇ 畫質可調性 (Quality Scalability)
不同層之間提供不同的畫面品質來做調節。

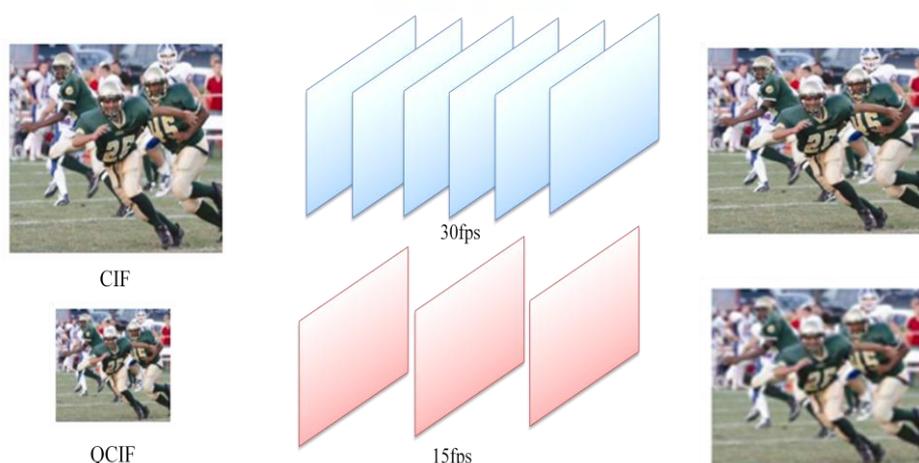


圖 3 可調視訊編碼所提供的三種不同調節性，由左至右分別為(1)空間可調性(2)時間可調性(3)畫質可調性。

2.1.2 層與層之間預測

在前一小節中說明可調視訊編碼提供的三種主要的可調性，但在不同層之間是利用什麼方法來達到可調節的好處呢，就是這小節所要介紹的層與層之間預測方法 (Inter-layer prediction)。在前一節中也有提到可調視訊編碼的基本架構是由基礎層和增進層所組成，增進層利用層與層之間預測的方法，盡可能的利用基礎層的資訊來達到減少編碼時所需的位元率。在可調視訊編碼中主要有三種層與層之間預測方法，包括有：

- ◇ 層與層之間的移動向量預測 (inter-layer motion prediction)
- ◇ 層與層之間的殘值預測 (inter-layer residual prediction)
- ◇ 層與層之間的畫面內預測 (inter-layer intra prediction)

因為位在增進層和基礎層相對位置的 MB (collocated MB) 在編碼資訊上擁有非常大的關係，基礎層的資訊若能適時且有效的給增進層在編碼時所使用，增進層就可以用最少的位元率來做編碼。接下來的小節中，將介紹三種層與層之間的預測。

2.1.2.1 層與層之間的移動向量預測

層與層之間的移動向量預測的做法如圖 4 所示，主要是增進層目前編碼的 MB 在做移動估測 (motion estimation) 時，利用基礎層相對位置 MB 的移動向量 (motion vector, MV) 做為預測移動向量 (motion vector predictor, MVP)，而且增進層和基礎層做移動估測時所使用的預測畫面是相同的。

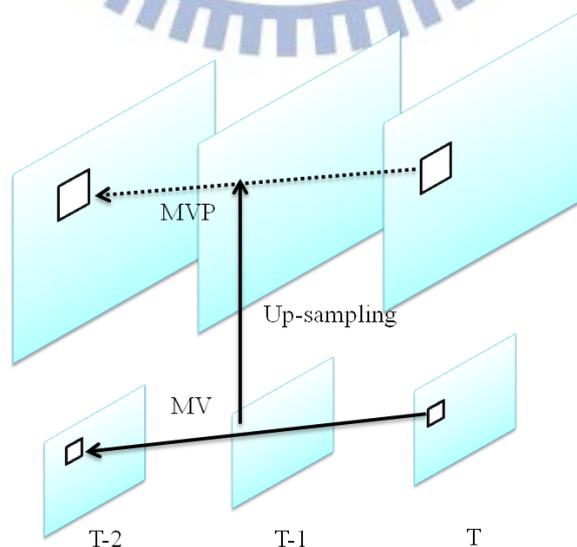


圖 4 層與層之間的移動向量預測示意圖

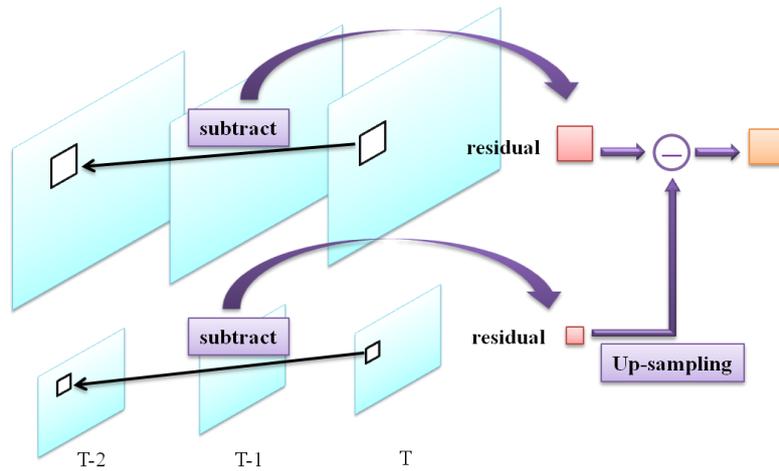


圖 5 層與層之間的殘值預測示意圖

2.1.2.2 層與層之間的殘值預測

如圖 5 所示，增進層在做完移動估測後，原始畫面的 Y 值會和預測畫面的 Y 值相減，此過程稱為移動補償（motion compensation）。在做完移動補償後會得到一差值，稱做為殘值（residual）。若沒有做層與層之間的殘值預測，則這個殘值就會在量化（quantization）完後直接去做編碼，但因為兩層之間的資訊關連度非常的高，兩層間的殘值亦可能非常的接近，因此增進層若有做層與層之間的殘值預測，增進層會將殘值和基礎層的殘值再做一次相減，使殘值能越少越好，盡可能減少編碼所需的位元率。

2.1.2.3 層與層之間的畫面內預測

增進層做層與層之間的畫面內預測需要有某種限制，此限制為基礎層相對位置的 MB 是用畫面內預測（intra prediction）做編碼。然而層與層之間的畫面內預測如圖 6 所示，基礎層相對位置的 MB 做完畫面內預測編碼且重建後，直接將此重建後的 MB 做為增進層相對位置 MB 的預測。

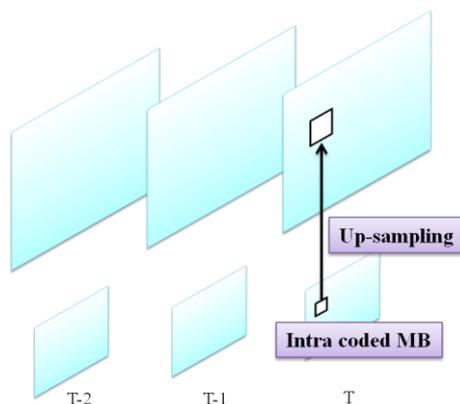


圖 6 層與層之間的畫面內預測

2.2 可調視訊編碼之編碼控制

在介紹完可調視訊編碼的三種主要的層與層之間的預測後，接下來將介紹可調視訊編碼的編碼控制。首先，介紹傳統的可調視訊編碼是做由下而上的編碼控制(Bottom-up Encoder Control, BUEC)，但是傳統的編碼控制在基礎層做編碼時，完全沒有考慮增進層如何去做編碼。此種做法和可調視訊編碼的基本想法—增進層盡可能使用基礎層的資訊做編碼的概念有點背道而馳。因為當基礎層在做編碼時，若完全沒有考慮增進層之後的編碼，增進層後來若想用基礎層的資訊做編碼，增進層變成只能附和基礎層的編碼去做決策。如此一來層與層之間的可調性就會變得沒這麼有效率。所以，接下來會介紹 Schwarz's *et al.* [5]所提出的多層編碼控制(Multi-layer Encoder Control, MLEC)，此種編碼控制是為了解決傳統編碼控制的問題。最後介紹由 Lin's *et al.* [6]所提出的多層編碼控制。

2.2.1 由下而上的編碼控制

傳統的可調視訊編碼在 JSVM[7]中所描述是使用由下而上的編碼控制做編碼決策，意思是先決定基礎層的編碼參數設定後，再去決定增進層的編碼參數設定。此編碼決策的方式以公式化表示為：

$$\begin{aligned} \min D_i(p_i | p_{i-1} \dots p_0) \\ \text{s.t.} \\ R_0(p_0) + R_1(p_1 | p_0) + \dots + R_i(p_i | p_{i-1} \dots p_0) \leq R_{ci} \end{aligned} \quad (3)$$

在公式(3)中， D_i 和 R_i 分別代表為第 i 層的失真(distortion)和位元率(rate)， p_i 是第 i 層設定的編碼參數， R_{ci} 則代表第 i 層的位元率最大限制。意思就是在位元率的限制下，如何去設定編碼參數能達到最小的失真，然而上層的編碼參數會被下層的編碼參數所影響。若以 widely-used Lagrangian approach 來表示兩層的編碼控制，首先基礎層的編碼參數決策表示為：

$$p_0^* = \arg \min_{\{p_0\}} D_0(p_0) + \lambda_0 \cdot R_0(p_0) \quad (4)$$

從公式(4)可以看出在傳統的編碼控制，當基礎層做編碼參數決策時完全不會考慮增進層的編碼參數設定，也就是盡可能的將基礎層的編碼做到最好。然而，增進層的編碼參數決策表示為：

$$\min_{\{p_1|p_0^*\}} D_1(p_1 | p_0^*) + \lambda_1 \cdot R_1(p_1 | p_0^*) \quad (5)$$

公式(5)顯示出增進層再做編碼參數決策時，會受到當初基礎層所決定的編碼參數而有所影響。因此，可以由公式(4)、(5)看出問題的所在，基礎層做編碼參數決策時只考慮基礎層單一層的結果，增進層會因為基礎層編碼參數的設定而可能產生較大的編碼失真，因為當初基礎層所選擇的編碼參數可能並不適合增進層所使用。基於這個理由，Schwarz's *et al.*[5]提出了一套權衡基礎層和增進層的方法，稱做多層編碼控制，而這個方法將在下一小節中詳細的介紹。

2.2.2 多層編碼控制

2.2.2.1 Schwarz's *et al.*所提出的多層編碼控制

為了解決傳統由下而上的編碼控制方法在編碼基礎層時只考慮單一層的缺點，Schwarz's *et al.*[5]提出了一種多層編碼控制的方法，這個方法是在編碼基礎層時也會同時考慮增進層。多層編碼控制在編碼基礎層時在基礎層和增進層中加入了一個權重(weighting)，利用這個權重決定在編碼時要以哪一層為主要考慮的編碼層。若以公式化表示基礎層的編碼決策為：

$$\begin{aligned} \min_{\{p_0, p_1|p_0\}} & (1-w) \cdot (D_0(p_0) + \lambda_0 \cdot R_0(p_0)) \\ & + w \cdot (D_1(p_1 | p_0) + \lambda_1 \cdot (R_0(p_0) + R_1(p_1 | p_0))) \end{aligned} \quad (6)$$

這公式(6)中是由公式(4)和公式(5)加上一個權重 w 演變而來，表示在兩層之間給予一個權重做權衡，此權重 $w \in [0; 1]$ 。在公式(6)中，若 $w = 0$ 代表基礎層在編碼決策時，回歸到傳統由下而上的編碼方法，只考慮基礎層的編碼結果。反之，當 $w = 1$ 代表基礎層編碼決策時，完全只考慮增進層的編碼結果，基礎層的編碼參數設定就變成只針對增進層去做最佳化選擇。然而在 Schwarz's *et al.*[5]所提出的多層編碼控制方法中，增進層在編碼決策時一樣是利用公式(5)去做決策。對於多層編碼控制決策的方法，也可以轉換成以下敘述的多目標最佳化問題：

$$\begin{aligned} \min & (1-w)D_0(p_0) + wD_1(p_1 | p_0) \\ \text{s.t.} & \\ (a) & (1-w) \times R_0(p_0) \leq R_{CB} \\ (b) & w \times (R_0(p_0) + R_1(p_1 | p_0)) \leq R_{CE} \end{aligned} \quad (7)$$

其中 R_{CB} 和 R_{CE} 分別是基礎層和增進層所限制的最大位元率。公式(7)中的式子(b)，在 $w = 0$ 時此位元率限制就不存在，但是在 $w = 1$ 時位元率限制就收斂到 R_{CE} ，因此公式(7)中的目標和限制會隨著權重 w 的不同而改變。但是此隨權重不同而目標和限制改變的現象，會造成位元率失真效能得不到預期的結果，因此 Lin's *et al.*[6]為了解決此問題而提出了改善方法，將在下一小節中提及。

2.2.2.2 Lin's *et al.*所提出的多層編碼控制

為了解決公式(7)中存在多目標限制會隨著權重 w 而改變的問題，Lin's *et al.*[6]提出了另一個多層編碼控制的多目標最佳化問題式子：

$$\begin{aligned}
 & \min (1-w)D_0(p_0) + wD_1(p_1 | p_0) \\
 & s.t. \\
 & (a) R_0(p_0) \leq R_{CB} \\
 & (b) R_0(p_0) + R_1(p_1 | p_0) \leq R_{CE}
 \end{aligned} \tag{8}$$

在公式(8)中(a)和(b)的限制已不受到權重的影響，但是權重 w 在於權衡基礎層和增進層方面依然是存在的。如此一來原先存在因改變權重 w 而改變目標限制的問題就不存在，也不會產生不可預期的位元率—失真效能。

然而，在 Lin's *et al.*[6]所提出的多層編碼控制方法中，最大的不同點在於基礎層和增進層在決策編碼參數時，只使用一個式子去做決策：

$$\begin{aligned}
 & \min_{\{p_0, p_1 | p_0\}} (1-w) \cdot D_0(p_0) + \lambda_0 \cdot R_0(p_0) \\
 & + w \cdot D_1(p_1 | p_0) + \lambda_1 \cdot (R_0(p_0) + R_1(p_1 | p_0))
 \end{aligned} \tag{9}$$

基礎層和增進層從公式(9)中就決定好兩層的編碼參數，這個想法是因為原先 Schwarz's *et al.* [5]所提出的方法在增進層是用公式(5)去做決策，但此舉像是多餘的。因為在公式(9)中，兩層的編碼參數應該就已經是確定的，增進層並不需要再做一次決策動作。

2.2.3 編碼控制應用於編碼模式決策

由於此篇論文是提出一套新的快速編碼模式決策演算法，因此在此小節中將利用編碼模式決策來介紹並比較兩種編碼控制方法。在此小節中都假設式應用在兩層

的可調視訊編碼架構上，且增進層和基礎層各有 N 種編碼模式需要做決策。先以傳統的由下而上的編碼控制為例：

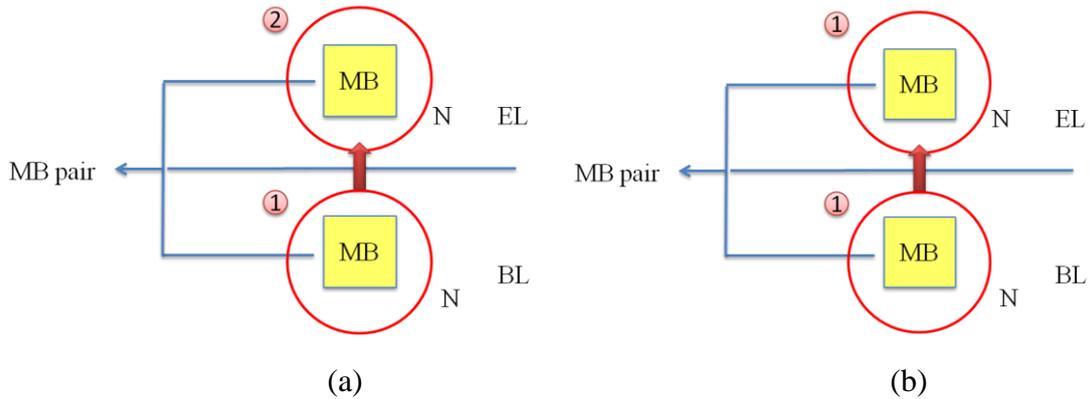


圖 7 (a)傳統由下而上編碼控制應用於編碼模式決策示意圖 (b) 多層編碼控制應用於編碼模式決策示意圖

如圖 7(a)所示，在不同層同一張畫面相對位置的 MB 稱做為一組 MB 配對(MB pair)，在傳統由下而上的編碼控制方法中，這組 MB 配對的編碼模式是先從基礎層做決策，決定完基礎層的編碼模式後再去決定增進層的編碼模式。若兩層各有 N 種編碼模式可以選擇，這種由下而上的編碼方式總共需要 $2N$ 次的編碼模式決策運算量。再來考慮多層編碼控制方法應用在編碼模式決策上，由圖 7(b)所示，這組 MB 配對的編碼模式是同時考慮的，若上下兩層都有 N 種編碼模式，在多層編碼控制方法中則需要 N^2 種編碼模式配對的決策運算量，這裡所說的編碼模式配對是上下兩層中各一個編碼模式可以形成一個配對，如圖 8 所示，若上下兩層各有 N 種編碼模式，則在基礎層的第一種編碼模式會和增進層的 N 種不同的編碼模式形成 N 種不同的編碼模式配對，因此在多層編碼控制時總共有 N^2 種編碼模式配對需要去考慮。

由上述所提到多層編碼控制做編碼模式決策時，若考慮所有的編碼模式進行決策則需要非常大的計算量，基於這個理由，作者接下來所設計的快速編碼模式決策演算法就是為了節省在多層編碼控制做編碼模式決策時所需的編碼時間，並且在位元率-失真效率方面盡可能的接近完整搜尋法。

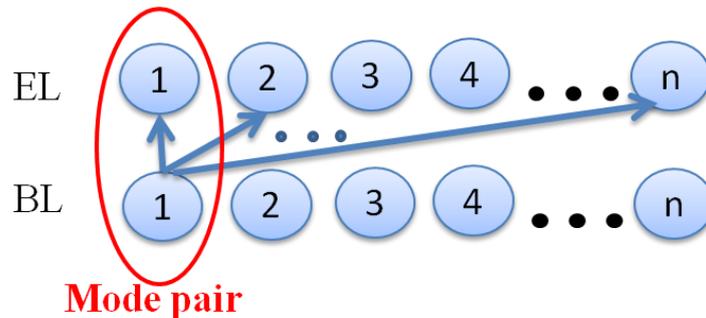


圖 8 編碼模式配對示意圖

第三章 快速編碼模式決策演算法

在第二章最後一小節提過，多層編碼控制在編碼模式決策時若考慮所有的編碼模式做決策需要花費非常大的計算量。因此為了提供一個更省時且有效率的編碼模式決策方法，作者在接下來的章節內將介紹一套新的快速編碼模式決策演算法，此方法是改善當初 Zhao's *et al.*[4]所提出演算法所存在的問題，並加上一套精進編碼模式決策的方法來提高位元率－失真的效能。

本篇論文所提出的快速編碼模式決策演算法是建立在可調視訊編碼的 CGS 架構且編碼控制方法是使用多層編碼控制方法。然而在此次演算法中加入編碼模式決策的編碼模式包括有，基礎層和增進層共有的 Skip、Inter16x16、Inter16x8、Inter8x16、Inter8x8、Intra4x4 和 Intra16x16，以及只有增進層獨有的 BLSkip 和 IntraBL，建成表格如下

表 1 測試的編碼模式種類表

編碼模式種類	基礎層(BL)	增進層(EL)
Inter MB Type	Skip Inter16x16 Inter16x8 Inter8x16 Inter8x8	Skip Inter16x16 Inter16x8 Inter8x16 Inter8x8
Intra MB Type	Intra16 Intra4	Intra16 Intra4
Inter-layer Prediction Type		BLSkip IntraBL

由此表看來基礎層有 7 種編碼模式需要決策，而增進層有 9 種。但是在增進層獨有的編碼模式 BLSkip 和 IntraBL 個別是當基礎層相對位置的 MB 是畫面間預測編碼(inter coded)和畫面內預測編碼(intra coded)時，增進層的 MB 才能選擇做此兩種編碼模式。因此實際上只有 56 組編碼模式配對須要考慮。為了接下來的章節能更清楚明白，在接下來的的方法中作者都是一次考慮一組編碼模式配對的 RD-cost 和編碼時間，以此為出發點設計快速編碼模式決策演算法。

3.1 編碼模式決策的兩項因素

在之前提過 Zhao's *et al.*[4]所提出的演算法純脆只是去計算下一個更好的編碼模式配對距離目前在編碼模式配對序列中停止的位置有多遠，但這個距離並不能有任何的保證，事實上並不是距離越遠所需編碼的時間就越多，距離越遠也不能保證有越多的 RD-cost 下降量，這兩者個別是由所需編碼的編碼模式配對種類以及位於編碼模式序列中的位置有所不同。原先 Zhao's *et al.*[4]所提出的辦法是去計算序列中每一個停止點位置的 y_k ，如公式(2)所示。但作者認為公式(2)中必須多考慮兩項因素，也就是不同位置的 RD-cost 下降量和不同編碼模式配對的運算時間。

然而編碼模式配對序列中不同位置的 RD-cost 下降量和編碼模式配對所需的運算時間這兩個因素可以將它們個別視為若停在這個位置之後若碰到更好的編碼模式配對可以得到多少“好處”和需要付出多少“代價”。因此作者所提出的演算法是去計算每一個編碼模式配對若下一個碰到更好的編碼模式配對能獲得多少的好處和需要付出多少代價，作者利用這兩者的比值稱做為“獲得好處與付出代價比”去找出在序列中的最佳停止點。以公式化來說明：

$$\begin{aligned} (a) & E[\Delta RD_k(T_k)] \\ (b) & E[C_k(T_k)] \\ (c) & y_k = \frac{E[\Delta RD_k(T_k)]}{E[C_k(T_k)]} \end{aligned} \tag{10}$$

公式(10)(a)是表示停在編碼模式配對序列中第 K 個位置的編碼模式，去計算之後若碰到更好的編碼模式配對能獲得多少期望 RD-cost 下降的好處。公式(10)(b)表示需要付出運算時間的代價。公式(10)(c)表示兩者的比值，也就是在第一章節有提到的 reward function，編碼模式配對序列中的每個位置，都可以去計算 reward function 並利用此去決定序列中最佳的停止位置。為了利用公式(10)去找出最佳停止位置，必須先去得到公式(10)中的(a)和(b)，因此接下來的兩小節就是去說明如何去獲得這兩種資訊。

3.1.1 考慮 RD-cost 下降量

個別討論如何加入這兩個因素，先討論編碼模式配對序列中不同停止位置，下一個遇到更好的編碼模式配對能獲得多少期望的 RD-cost 下降量 $-E(RD_{T_k})$ 是如何計算，將其公式展開為：

$$\begin{aligned}
 & E[\Delta RD_k(T_k)] \\
 &= \sum_{j=k+1}^N E[\Delta RD_k(T_k) | T_k = j] \cdot P(T_k = j) + 0 \cdot P(T_k = N + 1) \quad (11) \\
 &= \sum_{j=k+1}^N \Delta RDTable(j|k) \cdot \sum_{i=0}^k p_i \left(\frac{1}{\sum_{r=0}^{j-1} p_r} - \frac{1}{\sum_{r=0}^j p_r} \right)
 \end{aligned}$$

公式(11)中的機率 $P(T_k = j)$ 是沿用 Zhao's *et al.*[4]所提出的機率計算方式，計算停止點 K 之後遇到相對更好的編碼模式配對在不同位置的機率是多少。 $P(T_k = N + 1)$ 如前面章節所述是 K 位置之後，不會遇到更好編碼模式配對的機率，但因為此機率是代表 K 之後不會遇到更好的編碼模式配對，也就是 K 之後沒有更小的 RD-cost，所以此機率前面乘上一個 0。最後可以由公式(11)中看出，若想要得到 $E[\Delta RD_k(T_k)]$ 則必須要先有 $\Delta RDTable(j|k)$ 的資訊，此資訊是代表若停在序列中第 K 個位置，K 之後的編碼模式配對若比 K 位置之前最小的 RD-cost 更小，則這個 RD-cost 下降量有多少。為了得到這個資訊，作者進一步去統計並建成表格。

因為在原先的 Duration Problem 中，是去計算每個停止位置下一個遇到更好的編碼模式配對能獲得多少的 RD-cost 下降量，因此在統計資料時也必須照著這個想法去做統計，如表 2 就是去統計每一個停止位置下一個遇到更好的編碼模式配對能獲得多少 RD-cost 下降量。表 2 是利用五種不同的視訊序列—mobile、football、foreman、harbour 和 crew 在可調視訊編碼的 CGS 架構和利用多層編碼控制方法，基礎層的 QP=26 和增進層的 QP=22 且權重 W=0.5 的環境下做統計所得的平均 RD-cost 下降量。表 2 中顯示序列中有 13 個編碼模式配對，綠色數字(縱排)代表目前在序列中停止的位置，紅色數字(橫排)代表下一個遇到更好的編碼模式是在序列中的哪一個位置。表中的數字則代表 RD-cost 的下降量。有了這個 RD-cost 下降量的表，就可以輕易的代入公式(11)中去計算每一個停止位置，若之後會遇到更好的編碼模式配對能得到多少的預期 RD-cost 下降量。

表 2 RD-cost 下降量表

BLQP = 26 ELQP = 22												
W=0.5												
K	2	3	4	5	6	7	8	9	10	11	12	13
1	8631.31	769.07	567.58	432.33	362.01	251.48	242.95	246.25	257.64	212.02	179.4	169.24
2		972.92	566.3	465.48	400.75	273.85	262.65	257.14	234.13	209.17	180.35	175.12
3			563.44	482.07	399.3	289.53	258.84	235.59	225.23	200.43	183.68	168.89
4				476.11	389.45	291.88	248.64	233.65	211.77	187.89	177.82	167.59
5					381.22	287.13	238.67	221.96	201.46	189.62	177.16	170.63
6						281.34	234.62	211.02	191.46	188.84	174.28	167.55
7							224.68	197.66	183.55	182.48	173.91	165.29
8								195.4	178.17	178.65	170.27	161.29
9									173.91	175.71	166.27	164.29
10										171.81	163.06	164.99
11											162.38	164.67
12												165.45

此表可以明顯的看出在同一停止點 K 之後遇到更好的編碼模式配對所能得到的 RD-cost 下降量會隨著編碼模式配對在序列中位置的編號越大，下降量會越小。例如停止點 K=1，若下一個遇到更好的編碼模式配對是在位置 3，其 RD-cost 下降量是 769.07；而若下一個遇到更好的編碼模式配對是在位置 12 則 RD-cost 下降量是 179.4。明顯的，位置 3 所得到的 RD-cost 下降量較多。為什麼會有此種現象？因為上述提過表 2 的統計方式是去計算下一個遇到更好的編碼模式配對有多少 RD-cost 下降量，遇到更好的定義若以位置 12 為例，代表位置 12 以前的編碼模式配對都沒有更小的 RD-cost，也就是代表停止點 K 前面最小的 RD-cost 已經很小(代表 K 之前的編碼模式配對的 RD 效能已經很好)，因此才會在序列中後面的位置碰到更好的編碼模式配對。由此點說明表 2 的資訊是符合 duration problem 原先的定義。然而，作者在不同的 QP 設定和不同的權重下都有建立此表，因此能根據不同的設定去參考不同的表，讓此 RD-cost 下降量表能更具適應性。

3.1.2 考慮編碼模式配對的運算複雜度

考慮 RD-cost 的因素後，接下來繼續考慮各種不同編碼模式配對的運算時間複雜度。在原先 Zhao's *et al.*[4]所提出的方法中，是去計算下一個更好的編碼模式配對離序列中停止的點距離多遠。但作者認為不應該是去計算距離多遠，而應該是去計算下一個更好的編碼模式配對離序列中停止的位置之間所做過的編碼模式配對需要在付出多少期望的運算時間— $E[C_k(T_k)]$ 才對，若以公式來看：

$$\begin{aligned}
 E[C_k(T_k)] &= \sum_{j=k+1}^N E[C_k(T_k) | T_k = j] \cdot P(T_k = j) + \left(\sum_{i=k+1}^N c_i \right) \cdot P(T_k = N + 1) \quad (12) \\
 &= \sum_{j=k+1}^N \left(\sum_{i=k+1}^j c_i \right) \cdot P(T_k = j) + \left(\sum_{i=k+1}^N c_i \right) \cdot P(T_k = N + 1)
 \end{aligned}$$

(a) $\sum_{i=k+1}^j c_i$: 編碼模式配對的運算時間總和

公式 (12) 的機率計算方式和公式 (11) 的機率相同，一樣是使用 Zhao's *et al.*[4] 的方法。其中公式 (12) (a) 是計算目前停止位置為 K 且下一個遇到更好的編碼模式配對前，還需要花多少運算時間。以圖 9 為例，假如目前停止的位置是序列中的第一個位置，其編碼模式配對型態 (mode pair type) 是 A，若下一個相對更好的是在位置 4，也就是位置 1 之後需要再做第二個到第四個位置的編碼模式配對才會碰到相對更好的。因此公式 (12) (a) 就是去計算編碼模式配對的型態為 B、C 和 D 三種不同的編碼模式配對總共需要花費多少的運算時間。所以再計算 $E(C_{T_k})$ 前，必須先知道各種編碼模式配對的運算時間複雜度是多少。

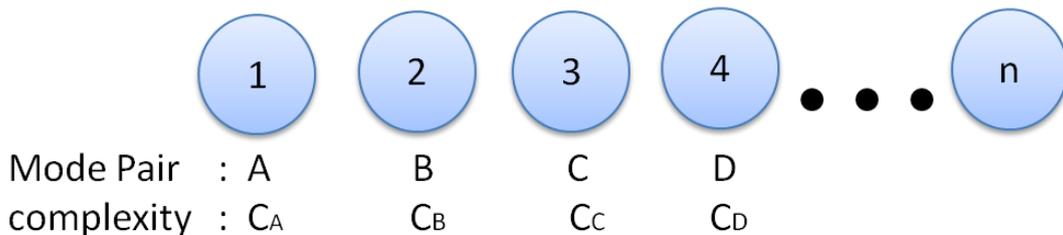


圖 9 序列中編碼模式配對的運算時間複雜度示意圖

表 3 各種編碼模式配對的運算時間表

1	8x8_8x8	2.95	21	Skip_8x8	1.52
2	8x8_8x16	2.85	22	Skip_8x16	1.44
3	8x16_8x8	2.85	23	8x8_intra16	1.41
4	8x8_16x8	2.82	24	Intra16_8x8	1.39
5	16x8_8x8	2.82	25	Skip_16x8	1.39
6	16x16_8x8	2.8	26	Skip_16x16	1.38
7	8x8_16x16	2.78	27	Intra16_8x16	1.34
8	8x16_8x16	2.75	28	Intra16_16x8	1.33
9	16x8_8x16	2.73	29	8x16_intra16	1.32
10	8x16_16x8	2.71	30	Intra4_8x8	1.31
11	16x16_8x16	2.71	31	8x8_intra4	1.31
12	8x16_16x16	2.7	32	Intra16_16x16	1.28
13	16x8_16x8	2.68	33	16x8_intra16	1.27
14	16x8_16x16	2.66	34	8x8_skip	1.26
15	16x16_16x8	2.65	35	8x16_intra4	1.23
16	16x16_16x16	2.65	36	Intra4_8x16	1.23
17	8x8_Blskip	2.5	37	Intra16_intra16	1.23
18	8x16_Blskip	2.37	38	8x16_skip	1.2
19	16x8_Blskip	2.35	39	16x8_intra4	1.2
20	16x16_Blskip	2.33	40	Intra4_16x8	1.2

41	16x16_intra4	1.19
42	Intra4_16x16	1.19
43	16x16_skip	1.18
44	Intra4_intra16	1.18
45	16x8_skip	1.17
46	Intra16_intraBL	1.15
47	Intra16_intra4	1.14
48	Skip_intra16	1.11
49	Intra16_skip	1.1
50	16x16_intra16	1.07
51	Intra4_intra4	1.06
52	Skip_Skip	1.05
53	Intra4_intraBL	1.02
54	Skip_Blskip	1.02
55	Intra4_skip	1
56	Skip_intra4	1

表 3 是利用五種不同的視訊序列編碼 100 張畫面統計獲得的，其統計方法為整個視訊序列只固定做某種編碼模式配對。以表 3 中編號 2 的編碼模式配對 8x8_8x16 為例，在整個視訊序列中基礎層 MB 的編碼模式只做 Inter8x8 且增進層只做 Inter8x16，然後去計算編碼時間，其他編碼模式配對的運算時間計算亦是如此。表 3 各個編碼模式配對後面的數字代表這個編碼模式配對的時間運算複雜度，數字越大代表其運算複雜度越大，這個運算複雜度係數是利用一個最小的運算時間當作基準和其他編碼模式配對的運算時間所形成的比值。

表 3 將運算複雜度的係數由大到小排序，可以由表中清楚的看出運算時間複雜度最大的編碼模式配對是 Inter8x8_Inter8x8，且 Inter8x8 的配對組合也都擁有較高的運算時間複雜度，這對編碼模式運算時間複雜度的認知是符合的，因為現實上 Inter8x8 是較細的 Inter 編碼模式，其的確需要較多的運算時間。因此在計算公式 (12) 的 $E[C_k(T_k)]$ 時，可以從編碼模式配對序列中知道要做過哪些編碼模式配對才會遇到一個更好的，這些要做的編碼模式都可以利用表 2 去做查表，找出相對應編碼模式配對的運算時間複雜度係數，如此一來就可以輕易的算出 $E[C_k(T_k)]$ 。

讀者可能會好奇的問，這種編碼模式配對的運算時間可能並不準確，因為這種測量方式是去計算整個編碼端的運算時間。作者在這邊的做法只是去給予初步的計算每種編碼模式配對的運算時間參考值，可能在運算時間上不是完全準確，但趨勢是正確的。因此作者的想法是，若未來能更準確的計算出每一個編碼模式配對的運算時間，這套演算法將能提供更準確的停止點。

3.1.3 同時考慮 RD 和運算複雜度，並找出停止點

接下來同時考慮 RD-cost 下降量和運算時間複雜度兩種因素，在前面有提到作者是將前兩小節所計算的期望值取一個比值 $y_k = E[\Delta RD_k(T_k)]/E[C_k(T_k)]$ 做為每一個停止位置的 reward function。這個值可以視為序列中的每一個停止點若之後會遇到相對更好的編碼模式配對能獲得多少 RD-cost 下降量的好處和需要付出多少運算時間的代價。然而這個 reward function 的趨勢是如何，且如何找到最佳的停止點呢？先來觀察 y_k 在序列中的趨勢如圖 10 所示：



圖 10 reward function— y_k 趨勢示意圖 BLQP=22, ELQP=18, W = 0.1

圖 10 中的縱座標是 y_k 值，橫坐標是序列中不同的停止點 $K=0\sim 11$ 。此實驗是在基礎層的 $QP = 22$ 、增進層的 $QP = 18$ 且權重 $w = 0.1$ 設定下統計的平均 y_k 值。可以從圖 10 中發現，不同的視訊序列的值都有相同的趨勢走向，都是嚴格遞減的趨勢。其中在 $K=1$ 以前的斜率非常的大， $K=1$ 之後就近趨平緩。

然而 y_k 值也可以視為若停在序列中的這個位置，若之後碰到更好的編碼模式配對能有多少的好處代價比。若 y_k 值近趨平緩，代表序列中後面的停止點所能得到的效益比已差不多，因此停在 y_k 值平緩的位置即可。但在平緩的位置上如何找到停止點呢？作者設計一個在不同的 QP 和不同的權重 w 設定下，規範一個規範值 (threshold)，也就是低於此規範值就是序列中的停止點。

此 y_k 的規範值設定方式是去實驗視訊序列在不同的規範值下，找出符合基礎層和增進層的位元率上升(bit rate increase)和峰值信噪比下降(PSNR decrease)分別在 2%和 0.05 dB 以內的值。如表 4 就是在不同設定下所找出 y_k 的規範值。

表 4 不同設定下 y_k 的規範值表

BLQP_ELQP	W01	W025	W05	W075	W09
22_18	1.5	1.3	1.1	1.3	0.9
26_22	2.5	2.7	1.9	1.9	1.5
30_26	3.7	3.9	3.5	3.1	2.1

然而表 4 中的規範值是個別在不同的設定下，利用不同的視訊序列得到的平均結果。以基礎層 QP = 22 和增進層 QP = 18 且權重 $w=0.1$ 的設定下舉例，做法如圖 11~圖 15 所示：

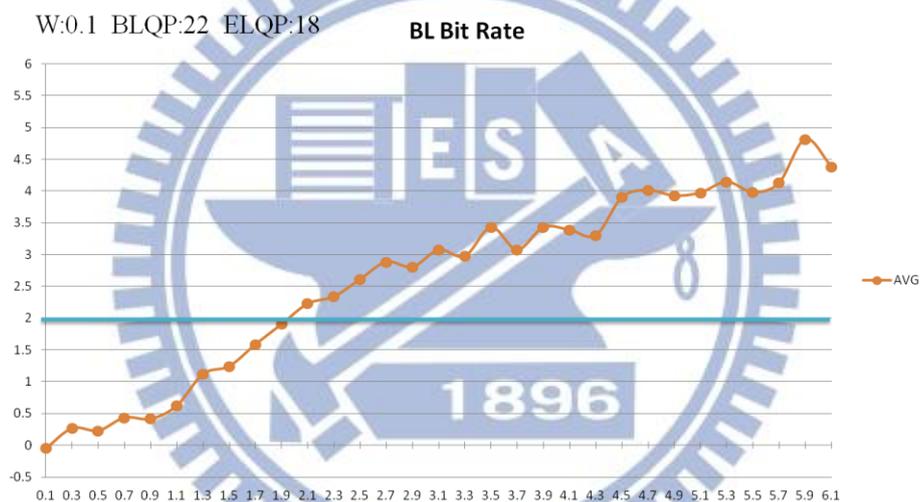


圖 11 基礎層在不同規範值的位元率上升圖

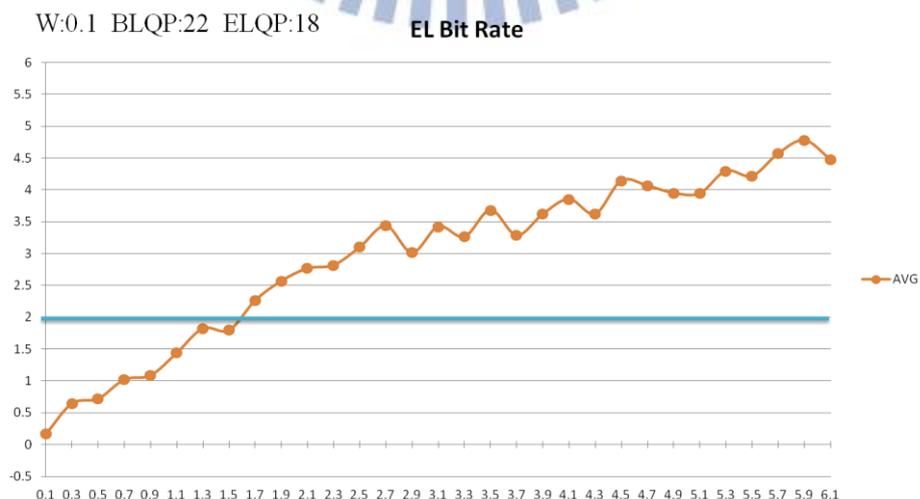


圖 12 增進層在不同規範值的位元率上升圖

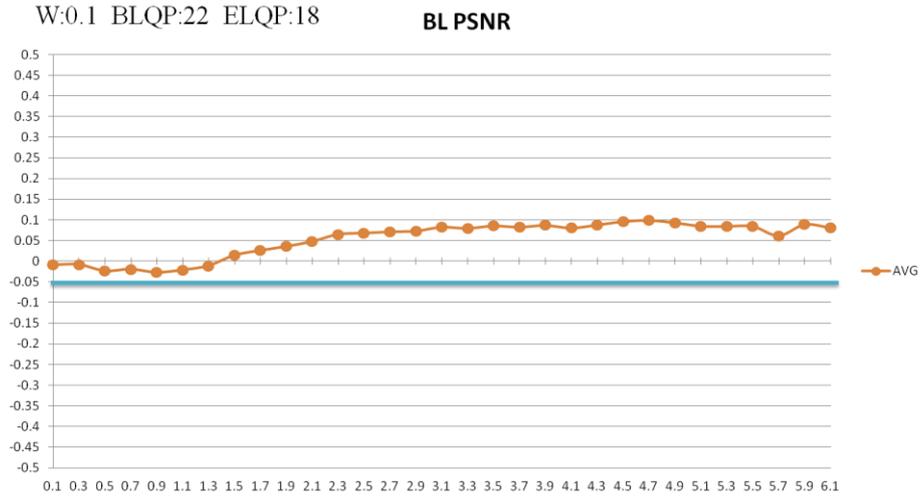


圖 13 基礎層在不同規範值的 PSNR 值



圖 14 增進層在不同規範值的 PSNR 值

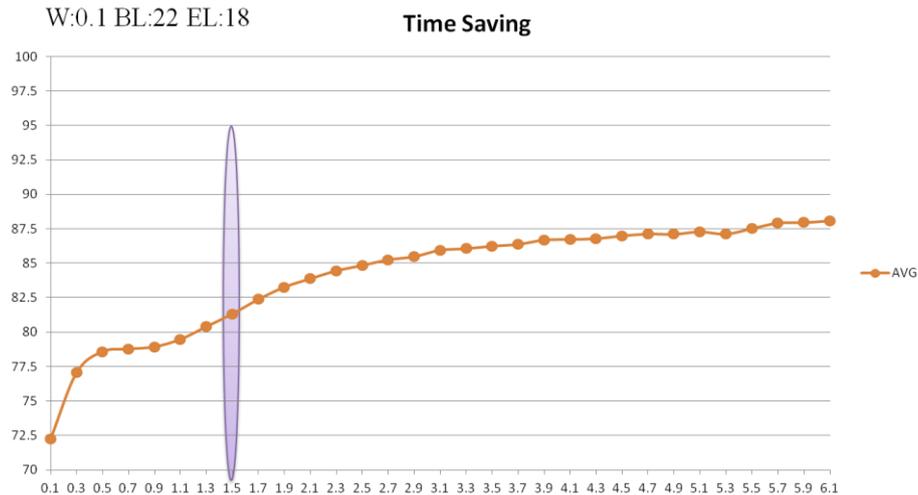


圖 15 在不同規範值的編碼時間節省

圖 11 和圖 12 分別是基礎層和增進層在不同規範值的編碼位元率上升圖，在此工作中是以上升在 2% 以內為限制並設定規範值。由圖中顯示基礎層和增進層都滿足此限制的值為 1.5。圖 13 和圖 14 亦是找出滿足峰值信噪比限制在 0.05 dB 最大下降量的點，圖中顯示基礎層和增進層在所有的規範值幾乎都滿足此種限制，所以聯集位元率上升和峰值信噪比下降的限制所找出的規範值就是 1.5。圖 15 中也顯示，在此組環境設定值下利用規範值 1.5 所得到的編碼時間節省將近有 81%。在其他的 QP 和權重 w 設定下，規範值的設定也是利用同樣的方法，作者所建議的規範值設定就是表 4 的結果。

當然 y_k 的規範值可以由使用者所設定，設定滿足使用者本身的需求。表 4 只是作者提出介於此種位元率—失真效能所建議的規範值。

3.2 編碼模式配對的序列產生

由於作者所提出的方法是在於改善 Zhao's *et al.*[4] 所提出的演算法，所以基於原本 Zhao's *et al.*[4]所提出的演算法，必須先產生一組編碼模式配對的序列。因此在此章節內將介紹編碼模式配對是如何產生的。

因為在參考論文[8]有提到，目前編碼的 MB 所選擇的編碼模式和周圍的 MB 或者和前一張畫面相對位置的 MB 所選擇的編碼模式具有相當大的相似度，也就是很有可能是用相同的編碼模式做編碼，但由於此次工作是在多層編碼控制方法的架構下設計快速演算法，因此是要一次考慮一組 MB 配對的編碼模式配對。基於這個理由，作者再進一步統計目前要編碼的 MB 配對其所選擇的編碼模式配對，和周圍的 MB 配對或前一張畫面相對應位置及周圍 MB 配對所選擇的編碼模式配對有多少機率會做到相同的編碼模式配對。如表 5 所示：

表 5 目前編碼的 MB 和周圍 MBs 做相同編碼模式配對的機率表

Sequence	3MBs	7MBs	11MBs	13MBs
Football	51%	67%	74%	76%
Mobile	48%	65%	72%	74%
harbour	57%	71%	76%	78%
Crew	55%	70%	76%	79%
Foreman	53%	68%	74%	76%
Akiyo	90%	93%	94%	94%
Avg.	59%	72%	78%	79%

表 5 是去統計目前編碼的 MB 配對和周圍的 MB 配對做相同編碼模式配對的機率有多少，也就是目前編碼的 MB 配對若和周圍其中一個 MB 配對做相同的編碼模式配對就會去累加次數。表中 3MBs、7MBs、11MBs 和 13MBs 分別是利用周圍多少 MB 配對的個數去做實驗。以圖 16 說明，若 (x, X) 是目前要編碼的 MB 配對， $(a, A) \sim (i, I)$ 是前一張畫面相對位置以及周圍的 9 個 MB 配對，且 $(j, J) \sim (m, M)$ 是同一張畫面周圍 4 個 MB 配對。表中 3MBs 代表是利用 (e, E) 、 (k, K) 和 (m, M) 這三組 MB 配對做實驗；7MBs 代表是多利用 (b, B) 、 (d, D) 、 (f, F) 和 (h, H) 這四組 MB 配對做實驗；11MBs 則是多利用 (a, A) 、 (c, C) 、 (g, G) 和 (i, I) 這四組 MB 配對；而 13MBs 則是利用 $(a, A) \sim (m, M)$ 這 13 組 MB 配對去做實驗。

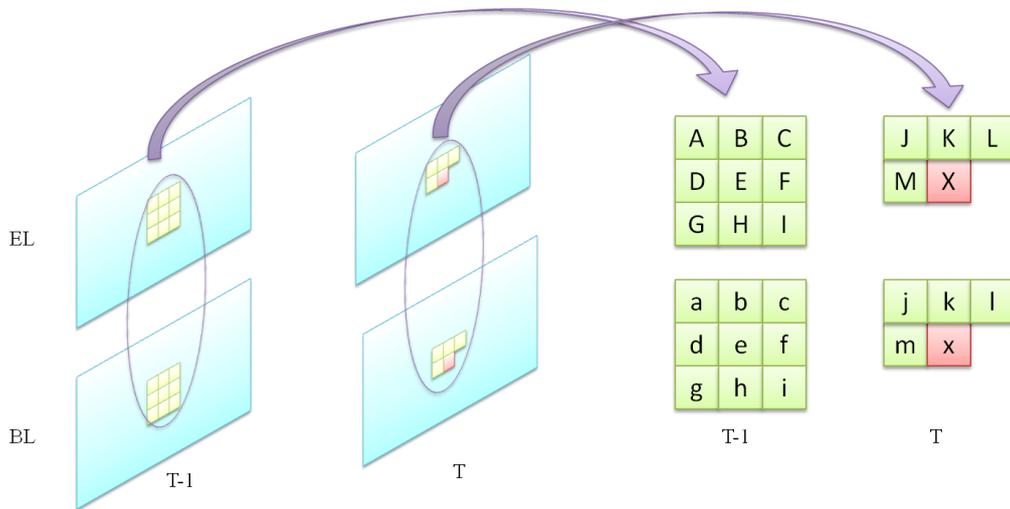


圖 16 利用哪些 MB 產生編碼模式配對序列示意圖

由表中可以清楚地看到，13 組 MBs 擁有將近 80% 的機率有其中一組 MB 配對和目前編碼的 MB 配對做相同的編碼模式配對。因此作者就利用這 13 個 MB 配對去產生一個編碼模式序列。

然而產生編碼模式序列的方式是先將周圍的 13 組 MB 配對所做的編碼模式配對形成一組機率，此機率是表示目前編碼的 MB 有多少可能會去做某種編碼模式配對。因為是用周圍的資訊產生的機率，所以亦稱做為區域機率 (local probability)。當然，若只有用區域機率產生編碼模式配對，由表 5 看來只會有將近 80% 的機率會和周圍做相同的編碼模式配對。但仍然還有其餘的 20% 不會和周圍的編碼模式配對相同。因此作者另外加入一組全域機率(global probability)，此全域機率是利用編碼不同的視訊序列，去統計出這些視訊序列做不同的編碼模式配對其機率各是多少。若用公式表示同時考慮兩組的機率：

$$\alpha \cdot local(P_i) + (1 - \alpha) \cdot global(P_i)$$

$$i = 1 \sim 56, \alpha = 0.7 \quad (13)$$

公式(13)中 P_i 代表是某一組編碼模式配對，在前面小節提到作者所考慮的編碼模式配對總共是 56 組，因此公式(13)也是考慮 56 組的機率。然而 α 設定為 0.7 是因為表 5 中顯示在大部分的視訊序列中，目前編碼的 MB 配對都有 75% 以上的機率可在周圍找到相同的編碼模式配對。最後利用公式(13)可以將 56 組編碼模式配對的機率算出。然而作者從 56 組編碼模式配對機率中，挑出前 13 大的機率組成一個編碼模式序列，其排序方式也是機率由大到小排列。然而每一張畫面最後一個 MB 配對做完編碼後，會對全域機率進行更新 (update)，讓全域機率能更符合不同的視訊序列以及環境。

3.3 編碼模式決策的精進方法

在上述的章節以介紹完如何利用區域和全域資訊產生編碼模式配對序列，且利用考慮 RD-cost 下降量和編碼模式配對的運算時間複雜度找出序列中的停止位置。但是若快速編碼模式決策演算法只做到上述就停止，將會有一些問題存在，問題包括有：

- ◇ 在相連畫面相對位置的 MB 幾乎只會產生相似的編碼模式配對序列。
- ◇ 在整個視訊序列中只有少數的編碼模式配對會被測試。
- ◇ 在位元率－失真效能方面會產生嚴重的失真結果。

為什麼會有這些問題存在？因為若你總是使用區域的資訊和全域的資訊去產生編碼模式配對序列，而且又給予區域的機率值較高的權重，會使得連續畫面相對位置 MB 所參考的資訊總是相似的，因此產生的編碼模式配對序列也就會非常相似，這種現象會一直存在，久而久之相對位置 MB 的編碼模式配對序列都會變成一樣，也就會形成總是只去測試相同的編碼模式配對。事實上不同畫面相對位置的 MB 還是有可能會去做差異非常大的編碼模式配對，因此若發生這種問題，編碼的位元率－失真效能將會有不可預期的失真產生。

為了解決上述問題，作者提出一個精進編碼模式決策的方法。為了方便起見，作者將之前章節所提出的辦法稱為第一步驟，然後精進的方法稱為第二步驟。在第一步驟中，是在編碼模式配對序列中找個停止點，也就 MB 做編碼模式決策是從序列的起始位置做到停止位置。因此作者所提出的精進方法就是在第一步驟做過的編碼模式配對中，其中最佳的編碼模式配對其 RD 效能還是不好，就必須跳脫原先方法去做一些其他種類的編碼模式配對。然而如何去判斷序列中最佳的編碼模式配對其 RD 效能好壞與否，而且是要做哪些其他種類的編碼模式配對，這第二步驟的方法將在之後詳細說明。

3.3.1 判定 RD 效能

作者所提出判定序列中最佳的編碼模式配對其 RD 效能好壞與否的方法是利用周圍 MB 配對的平均 RD 做比較，此想法在參考論文[9]~[13]也有被提到，都是利用周圍 MB 的平均 RD 或是整張畫面的平均 RD 行為做為參考，用來比較目前編碼 MB 的 RD 效能好壞與否。至於利用哪些周圍的 MB 配對比較 RD 如圖 16 所示，利用已編碼完的 (j, J) 、 (k, K) 、 (l, L) 和 (m, M) 四組同一張畫面 MB 配對的平均 RD 和目前編碼 MB 配對的最佳 RD 取一個比值，如公式(14)所示：

$$RDratio = \frac{RD(MB_{current})}{RD(MB_{Neighbor})} \quad (14)$$

公式(14)中 $RD(MB_{current})$ 是表示目前編碼的 MB 配對在第一步驟中所找到最佳編碼模式配對的 RD-cost。 $RD(MB_{Neighbor})$ 則是代表周圍 4 個 MB 的平均 RD-cost，作者接下來對這個 RD 比值進一步分析與討論。作者將前一張畫面相對位置 MB pair 的 RD 比值和目前編碼 MB pair 的 RD 比值做了一項實驗觀察，觀察兩個 MB pair 的 RD 比值之間的關係，表 6 顯示實驗結果。

表 6 目前編碼 MB pair 和前一張畫面相對位置 MB pair 的 RD ratio 差值

Ratio(Current MB pair) - Ratio(Collocated MB pair) < 0.3							
22_18	mobile	football	harbour	foreman	crew	akiyo	Avg
W01	0.95	0.91	0.97	0.89	0.89	0.95	0.93
W025	0.93	0.91	0.96	0.89	0.87	0.94	0.92
W05	0.89	0.91	0.93	0.88	0.87	0.93	0.9
W075	0.83	0.9	0.87	0.84	0.82	0.93	0.87
W09	0.79	0.88	0.83	0.81	0.82	0.94	0.84
26_22	mobile	football	harbour	foreman	crew	akiyo	Avg
W01	0.9	0.91	0.93	0.9	0.87	0.96	0.91
W025	0.88	0.9	0.9	0.89	0.85	0.95	0.9
W05	0.86	0.9	0.89	0.88	0.85	0.95	0.89
W075	0.83	0.89	0.86	0.85	0.83	0.94	0.87
W09	0.8	0.85	0.8	0.81	0.86	0.95	0.84
30_26	mobile	football	harbour	foreman	crew	akiyo	Avg
W01	0.88	0.89	0.91	0.89	0.86	0.97	0.9
W025	0.88	0.89	0.89	0.9	0.84	0.97	0.89
W05	0.86	0.88	0.89	0.88	0.84	0.96	0.88
W075	0.84	0.86	0.86	0.85	0.84	0.96	0.87
W09	0.81	0.83	0.83	0.83	0.9	0.96	0.86

表 6 統計前後畫面相對位置的 MB pair 其 RD 比值的差，有多少比例是在小於 0.3 的情況。將近 90% 的機率顯示兩個相對位置的 MB pair 的 RD ratio 的差值小於 0.3。因此作者利用這個結果來判斷目前編碼的 MB pair 在第一步驟獲得的 RD-cost 好壞與否。

3.3.2 精進的方法

上一小節說明如何去判斷在第一步驟中所做的最佳編碼模式配對，其 RD 效能的好壞與否。接下來的小節將介紹，如何對不好的 RD 效能進一步去做精進的程序。作者所設計的精進方法是分兩種階級的，限定兩種等級的 RD ratio 差值 -0.3 和 0.5。也就是 RD ratio 差值若超過 0.3 和 0.5 分別會有不同的精進方法。以公式來說明：

$$(a) \ 0.3 + RDratio_{previous_frame} \leq RDratio_{current_frame} < 0.5 + RDratio_{previous_frame} \quad (15)$$
$$(b) \ RDratio_{current_frame} \geq 0.5 + RDratio_{previous_frame}$$

公式(15)中的(a)和(b)代表兩種不同級別的 RD ratio 值，簡稱為級別(a)和級別(b)。級別(a)代表目前編碼的 MB pair 在第一步驟中獲得的 RD 比周圍的平均差，但差異較小，因此精進的方法就沒這麼複雜。然而級別(b)則代表差異較大，需要做較複雜的精進步驟。級別(a)的精進方法為：

1. 在快速演算法的第一步驟中所找到最佳的編碼模式配對，根據不同的權重 w ，固定此最佳編碼模式配對中某一層的編碼模式，並在另外一層測試不同的編碼模式。
2. 權重 $w \leq 0.5$ 時，固定基礎層的編碼模式；權重 $w > 0.5$ 時，固定增進層的編碼模式。另外一層所測試的編碼模式以表 7 和表 8 做說明，表 7 和表 8 分別是固定基礎層和固定增進層後，依序由左至右測試不同的編碼模式，此順序是經由視訊序列統計在固定某一層的編碼模式後，另外一層所做的編碼模式依照機率大小由左至右排序。
3. 依序測試完表 7 和表 8 中另一層的某一個編碼模式後，就會進行評估此編碼模式配對的 RD 是否符合周圍的平均 RD。測試方式以公式(16)做判斷，

$$Ratio_{current_frame} - Ratio_{previous_frame} < 0.3 \quad (16)$$

4. 若符合公式(16)，則結束此次編碼模式決策。否則繼續測試表 7 或表 8 中的編碼模式。

然而級別(b)精進方法，是在做級別(a)的方法前根據不同的權重 w ，個別先去測試表 7 和表 8 中固定某一層的編碼模式，另外一層做表中紅色部份的編碼模式。

在級別(b)中已經不考慮序列中的最佳編碼模式配對是什麼，原因是 RD ratio 超出規範值太多，認為序列中最佳的編碼模式所得的 RD 效益太差，需要完全的跳出序列中原先的選擇。以表 7 的 $w=0.1$ 為例，在級別(b)中會先測試 skip_skip，inter16x16_skip、inter16x8_skip、inter8x16_skip、inter8x8_skip、intra4_skip 和 intra16_skip 這七組編碼模式配對。在七組中挑出一組最佳 RD 的編碼模式配對並根據權重 w 固定某一層的編碼模式，對另外一層的編碼模式依序測試不同的編碼模式，其做法如級別(a)的精進方法。

表 7 精進方法的編碼模式配對順序表－固定基礎層

W=0.1	EL							
BL								
skip	skip	inter16x16	inter8x8	inter8x16	inter16x8	Blskip	intra16	intra4
inter16x16	skip	Blskip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
inter16x8	skip	Blskip	inter16x16	inter16x8	inter8x16	inter8x8	intra16	intra4
inter8x16	skip	Blskip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
inter8x8	skip	Blskip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
intra4	skip	intraBL	inter16x16	inter8x16	inter16x8	inter8x8	intra4	intra16
intra16	skip	intraBL	inter16x16	inter8x16	inter16x8	inter8x8	intra4	intra16
W=0.25	EL							
BL								
skip	skip	inter16x16	inter8x16	inter8x8	inter16x8	Blskip	intra16	intra4
inter16x16	Blskip	skip	inter8x16	inter16x16	inter16x8	inter8x8	intra16	intra4
inter16x8	Blskip	skip	inter16x16	inter16x8	inter8x16	inter8x8	intra16	intra4
inter8x16	Blskip	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
inter8x8	Blskip	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
intra4	skip	intraBL	inter16x16	inter8x16	inter16x8	inter8x8	intra4	intra16
intra16	skip	intraBL	inter16x16	inter8x16	inter16x8	inter8x8	intra4	intra16
W=0.5	EL							
BL								
skip	skip	inter16x16	inter8x16	inter8x8	inter16x8	Blskip	intra16	intra4
inter16x16	Blskip	skip	inter8x16	inter16x8	inter16x16	inter8x8	intra16	intra4
inter16x8	Blskip	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
inter8x16	Blskip	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
inter8x8	Blskip	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
intra4	intraBL	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra4	intra16
intra16	intraBL	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra4	intra16

表 8 精進方法的編碼模式配對順序表—固定增進層

W=0.75, W=0.9	BL						
EL							
skip	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
inter16x16	skip	inter16x16	inter8x16	inter16x8	inter8x8	intra16	intra4
inter16x8	skip	inter16x16	inter16x8	inter8x16	inter8x8	intra16	intra4
inter8x16	skip	inter16x16	inter8x16	inter8x8	inter16x8	intra16	intra4
inter8x8	skip	inter16x16	inter8x8	inter8x16	inter16x8	intra16	intra4
BLSkip	skip	inter16x16	inter8x16	inter8x8	inter16x8		
intra4	skip	inter16x16	inter8x16	inter8x8	inter16x8	intra16	intra4
intra16	skip	inter16x16	inter8x16	inter8x8	inter16x8	intra16	intra4
intraBL	intra16	intra4					

最後在此說明為什麼以權重 $w \leq 0.5$ 和 $w > 0.5$ 為分界點，分別先固定基礎層或增進層的編碼模式。因為在前面介紹過多層編碼控制是使用權重 w 來給予層與層之間做調節，權重小的時候代表是給予基礎層較大的權重，也就是編碼參數設定以基礎層為考量；反之編碼參數設定是以增進層為考量。基於這個理由，作者認為在權重 $w \leq 0.5$ 時，編碼參數的設定是以基礎層為考慮多一點，因此認定為基礎層當初選擇的編碼模式是較準確的，因此固定基礎層的編碼模式，並嘗試增進層的其他不同的編碼模式，反之在權重 $w > 0.5$ 時則剛好相反。

3.3.3 精進程序前後差異比較

為了證實精進程序對於作者所提快速演算法的重要性，在此小節提出實驗證明。此實驗分別在有執行精進方法和沒有執行精進方法中，個別去統計有多少編碼模式配對的種類會在序列中被測試。表 9 顯示實驗結果，

表 9 序列中編碼模式配對種類的比較

	No Refine					Refine			
mobile	22_18	26_22	30_26	34_30		22_18	26_22	30_26	34_30
W01	13	13	14	14		24	29	30	30
W025	8	15	16	16		14	30	29	29
W05	10	14	16	16		27	26	28	29
W075	13	13	16	14		24	26	27	27
W09	13	14	17	15		26	27	26	23
AVG	11.4	13.8	15.8	15		23	27.6	28	27.6

football									
W01	10	14	16	19		21	25	26	28
W025	8	15	17	16		18	18	24	25
W05	8	17	15	19		18	24	26	29
W075	11	15	16	18		18	21	27	24
W09	13	14	14	16		17	21	25	21
AVG	10	15	15.6	17.6		18.4	21.8	25.6	25.4
harbour									
W01	11	13	13	14		23	20	26	27
W025	10	14	13	14		15	24	26	26
W05	7	13	13	13		21	27	23	27
W075	10	13	13	14		17	22	25	24
W09	14	13	14	15		22	28	25	18
AVG	10.4	13.2	13.2	14		19.6	24.2	25	24.4
foreman									
W01	16	17	16	16		30	29	28	30
W025	16	17	20	15		28	28	30	27
W05	19	16	14	15		29	27	29	26
W075	15	16	14	13		29	29	29	27
W09	15	15	14	16		29	29	28	21
AVG	16.2	16.2	15.6	15		29	28.4	28.8	26.2
crew									
W01	14	18	17	15		26	32	32	30
W025	14	14	16	18		26	30	30	26
W05	13	15	16	17		30	30	32	30
W075	14	14	18	16		29	30	27	22
W09	16	16	17	17		34	21	18	19
AVG	14.2	15.4	16.8	16.6		29	28.6	27.8	25.4

此實驗是在不同的視訊序列在不同的 QP 和不同的權重 w 下測試，NoRefine 顯示沒有做精進方法時，序列中的編碼模式配對種類個數；Refine 則是顯示有做精進方法，時序列中的編碼模式配對種類個數。由表 9 中可以發現有做精進方法的序列中，其編碼模式配對的種類會多出很多，此實驗結果也能證明此章節開頭所述若沒做精進程序會形成的問題。因此精進方法在作者所提出的快速演算法中是非常重要的環。

3.4 快速編碼模式決策演算法的流程

在介紹完如何做快速編碼模式決策演算法後，其步驟利用條列的方式呈現，一步步介紹演算法的完整流程。如圖 17 所示，圖中左半部的藍色正方形是目前編碼的 MB pair，右半部是編碼模式配對序列。

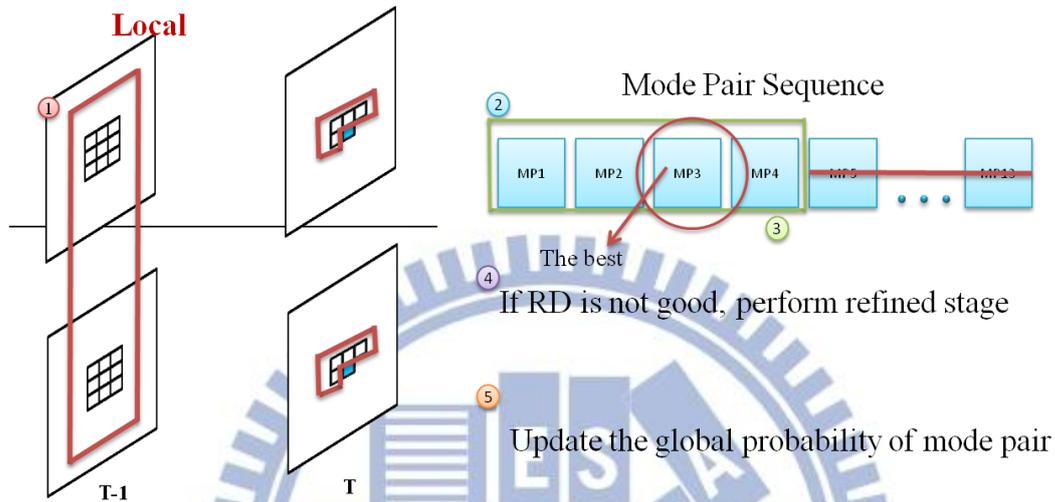


圖 17 演算法流程步驟圖

Step1

利用周圍的 MB pair 和前一張畫面相對位置以及周圍的 MB pair 的資訊，產生出一組區域的編碼模式配對機率。(詳細做法在 3.2 說明)

Step2

利用編碼模式配對的區域機率和全域機率形成一個編碼模式配對的序列。(詳細做法在 3.2 說明)

Step3

決定編碼模式配對序列中的停止點，依序測試序列中停止位置前的所有編碼模式配對並找出一組最小 RD-cost 的編碼模式配對。(詳細做法在 3.1 說明)

Step4

將 step3 所得到的 RD-cost 和周圍平均的 RD-cost 取個 RD ratio，並和前一張畫面相對位置的 RD ratio 比較，若超出規範值則進行精進的程序(精進程序分為兩個級別)。(詳細做法在 3.3 說明)

Step5

編碼完一張畫面中最後一個 MB pair 後，對編碼模式配對的機率進行更新。

第四章 實驗結果與討論

本篇論文所提出的快速編碼模式決策演算法，在此章節內將和不同的演算法比較 RD—performance 和編碼時間，其中包括完整搜尋法、Zhao's *et al.*[4]和 Lin's *et al.*[6]所提出的快速演算法。

4.1 實驗環境

此次工作所提出的快速編碼模式演算法、完整搜尋法、Zhao's *et al.*[4]和 Lin's *et al.*[6]所提出的快速演算法，皆在 JSVM 9.12.2 [7]中實做並測試。且所有的實驗皆在 CPU: Intel Q9650 4GHz，RAM：8G 的 PC 上執行，表 10 顯示所有測試的架構和設定。

表 10 實驗環境設定

Prediction structure	IPPP
Inter-layer coding structures	CGS
Resolution: frame per second	QCIF : 30
Frame number	31
QP	Fixed QP setting $QP_E = 18, 22, 26$ $QP_B = QP_E + 4$
Motion Estimation	MLEC
Inter-Layer Residual Prediction	available

4.2 實驗比較與討論

本實驗分別在不同視訊序列和不同的量化參數(QP)和權重(w)設定下，分別去比較不同演算法的位元率－失真效能(rate-distortion performance)。比較方法是不同的演算法和完整搜尋法(exhaustive search)比較並計算元率上升(bit rate increase)和峰值信噪比下降(PSNR decrease)以及編碼時間節省(time saving)，再將這些值拿出來比較效能。其中位元率上升、峰值信噪比下降以及編碼時間節省的計算方式如下：

$$\Delta \text{BitRate} = \frac{(\text{快速演算法的 Rate} - \text{完整搜尋的 Rate})}{\text{完整搜尋的 Rate}} \times 100\%$$

$$\Delta \text{PSNR} = (\text{快速演算法的 Y-PSNR}) - (\text{完整搜尋法的 Y-PSNR})$$

$$\text{TimeSaving} = \frac{(\text{完整搜尋法的編碼時間} - \text{快速演算法的編碼時間})}{\text{完整搜尋法的編碼時間}} \times 100\%$$

在本次的工作中主要去做 3 種不同的實驗 (1) 和 Zhao's *et al.*[4]所提出的演算法做比較，因為本論文所提出的方法是多考慮兩個因素－不同位置的 RD-cost 下降量以及不同編碼模式配對的運算複雜度 (2) 將 Zhao's *et al.*[4]的演算法加入本論文所提供的精進方法，比較有無做精進方法的實驗結果。(3) 在其他非統計視訊序列中是否有一樣好的 RD 效能，這裡純粹和完整搜尋法比較。

4.2.1 實驗一

在實驗一中提出三個演算法在不同 QP 設定以及不同權重 w 之間的比較，在實驗數據中可以發現，作者所提出的演算法編碼時間節省是介在其他兩種演算法之間，但在兩層之間的 $\Delta \text{Bitrate}$ 和 ΔPSNR 是三種演算法中最穩定且效能也最好的。至於編碼時間節省為什麼會比 Zhao's *et al.*[4]所提的演算法少，因為 Zhao's *et al.*[4]的演算法沒有考慮 RD-cost 和不同編碼配對的運算複雜度，因此在編碼模配對序列中停止點不是停在最佳的停止位置，往往會停在序列中的前半部，後面的編碼模式配對就很少考慮到，因此編碼時間節省會較多，但其 RD 效能方面就顯得較差。然而 Lin's *et al.*[6]所提出的辦法中，編碼模式決策的步驟都很固定，往往都是先測試相同的編碼模式配對，因此較不能適應不同的視訊序列，因此在編碼效能方面顯得較差。

表 11 BLQP = 22 ; ELQP = 18 不同演算法的比較

Sequence	Weighting	Lin's					Zhao's					Proposed				
		TS	BL		EL		TS	BL		EL		TS	BL		EL	
		TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR
	W01	81.08	-0.02	-0.09	-0.04	0.41	85.44	0.08	1.3	0.38	2.1	83	0.11	1.86	0.08	2.2
	W025	81.97	-0.12	-1.28	-0.22	-0.91	86	-0.03	0.42	0.01	0.26	83.05	0.11	2.02	0.26	2
mobile	W05	70.18	0.08	0.58	0.02	0.05	85.65	-0.05	0.43	0.34	2.09	82.91	0.02	0.87	0.19	1.38
	W075	72.23	-0.01	0.14	0.01	-0.01	85.44	0.02	0.39	0.06	1.31	82.31	0.08	1.54	0.04	1.34
	W09	73.24	-0.19	2.16	-0.05	-0.96	86.9	-0.25	4.27	0.02	0.91	80.35	-0.03	2.2	0.01	0.41
	W01	79.33	-0.03	0.09	0.05	0.78	86.26	-0.12	0.97	0.77	1.66	80.98	-0.06	0.83	0.63	1.8
	W025	79.31	-0.05	0.02	-0.04	0.6	86.24	-0.12	1.69	-0.02	2.01	81.41	-0.1	1.87	-0.01	2.24
football	W05	62.98	-0.1	0.09	-0.05	0.48	85.68	-0.06	2.27	-0.12	1.99	84.29	-0.05	2.27	-0.04	2.09
	W075	69.61	0.09	0.38	-0.06	0.41	85.47	0.42	2.9	-0.15	1.55	82.1	0.25	2.43	-0.05	1.54
	W09	71.14	-0.04	0.54	-0.05	0.63	84.68	-0.22	1.72	-0.09	2.52	78.28	-0.07	1.24	-0.02	1.45
	W01	76.97	-0.02	0.08	0.04	0.82	84.83	0.07	1.6	0.19	1.99	83.78	0.07	1.57	0.23	1.97
	W025	83.98	-0.1	-0.66	-0.15	-0.6	87.38	0.26	3.62	0.54	4.22	84.94	0.04	1.51	0.16	1.48
harbour	W05	76.87	0.04	0.43	0.01	-0.08	87.51	0.03	1.5	0.06	1.03	85.61	-0.04	0.82	0.06	0.89
	W075	78.2	-0.01	-0.2	-0.01	-0.25	87.28	0.05	0.48	-0.01	0.2	83.77	-0.01	0.58	-0.02	0.55
	W09	75.31	0	3.05	-0.02	-0.53	83.39	0.04	0.46	-0.01	-0.46	81.29	0.1	1.95	0.02	0.28
	W01	79.71	0	0.57	-0.11	1.08	85.31	-0.03	1.56	-0.1	3.49	79.22	-0.03	1.14	-0.04	1.55
	W025	80.27	-0.02	0.97	-0.1	2.55	85.99	-0.05	0.83	0.45	1.44	81.82	-0.03	1.07	0.25	2.12
foreman	W05	71.83	0.06	1.14	0.02	0.58	85.52	0.39	4.83	-0.48	-2.75	82.37	0.16	3	-0.05	-0.59
	W075	73.48	-0.05	1.37	-0.01	-0.33	84.27	-0.29	0.84	0.01	2.08	80.26	-0.09	2.38	0.02	2.53
	W09	75.17	-0.19	4.8	0.01	-1.68	87.19	-0.89	13.41	0.05	5.49	79.06	-0.04	2.15	0.04	0.07
	W01	81.26	-0.02	0.04	-0.13	0.69	88.78	-0.05	1.92	-0.13	3.43	79.69	-0.02	0.83	-0.03	1.47
	W025	82.15	-0.15	-0.39	-0.16	1.39	88.44	-0.13	2.33	0.24	3.02	80.6	-0.06	1.45	0.26	1.83
crew	W05	69.21	-0.09	5.07	-0.19	3.33	87.98	0.48	6.3	-0.58	-1.78	81.5	0.27	2.88	-0.04	1.08
	W075	72.95	-0.17	4.12	-0.16	2.89	88.66	-0.22	1.84	0.05	3.92	81.12	-0.02	2.54	0.09	2.83
	W09	77.1	-0.23	6.18	-0.1	1.18	93.04	-0.7	7.03	0.08	0.52	84.67	-0.04	2.33	0.06	-0.31
Average		75.82	-0.05	1.17	-0.06	0.5	86.53	-0.05	2.6	0.06	1.69	81.94	0.02	1.73	0.08	1.37

表 12 BLQP = 26 ; ELQP = 22 不同演算法的比較

Sequence	Weighting	Lin's					Zhao's					Proposed				
		TS	BL		EL		TS	BL		EL		TS	BL		EL	
		TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR
	W01	81.93	-0.06	-0.42	-0.14	-0.12	87.65	-0.04	0.69	-0.26	4.04	83.82	-0.02	0.78	-0.02	1.23
	W025	83.3	-0.07	-0.5	-0.22	0.3	85.87	-0.04	0.21	0.33	-0.3	83.06	-0.01	0.95	0.31	0.45
mobile	W05	71.94	0.04	1.39	0.08	0.42	84.62	0.13	2.93	-0.03	0.41	79.75	0.05	1.88	0	0.44
	W075	73.51	-0.21	0.08	0.09	0.81	86.98	-0.17	3.67	0.39	4.94	79.32	-0.05	2.15	0.24	3.01
	W09	75	-0.25	7.41	-0.04	-1.44	87.44	-0.93	14.97	0.29	4.22	79.88	-0.07	2.12	0.17	1.23
	W01	82.23	-0.04	-0.05	-0.02	1.11	86.03	-0.11	0.63	1.3	2.09	82.62	-0.04	0.33	0.25	1.89
	W025	81.54	-0.02	0.08	-0.03	0.83	86.13	-0.09	1.52	0.01	2.23	82.19	-0.06	1.83	0.06	2.71
football	W05	68.72	-0.07	0.34	-0.05	0.49	85.76	0.01	2.57	-0.14	1.99	80.9	0.04	2.66	-0.1	2.26
	W075	69.96	0.09	0.47	-0.06	0.26	85.43	0.78	4.17	-0.15	1.32	80.06	0.05	1.67	-0.05	1.77
	W09	71.97	0.2	1.08	-0.08	1.1	84.98	-0.36	1.04	-0.11	3.39	80.6	-0.01	0.83	0	0.54
	W01	81.5	-0.02	0.36	-0.13	0.15	84.64	-0.07	-0.14	-0.02	0.88	86.83	-0.03	0.83	-0.03	1.1
	W025	82.4	-0.09	-0.47	-0.17	-0.07	83.97	-0.07	-0.06	0.19	0.29	83.8	-0.03	0.92	0.32	1.26
harbour	W05	72.5	0.05	0.87	0.02	0.12	82.49	-0.05	1	0.03	1.32	81.66	0.06	1.89	0.04	1.29
	W075	74.66	0.01	1.43	-0.01	-0.1	82.51	-0.05	-0.22	0.02	0.43	84.66	-0.04	1.71	0.03	1.32
	W09	76.18	-0.25	7.12	-0.04	-1.79	83.26	-0.18	7.8	0	-1.67	81.97	-0.04	2.03	0.07	0.13
	W01	82.55	0.01	0.67	0.01	2.13	87.47	0.04	3.65	-0.8	6.9	84.82	0.01	2.64	-0.03	3.86
	W025	82.19	-0.07	0.03	-0.13	2.26	86	-0.13	0.08	0.48	1.73	83.56	-0.04	1.4	0.28	3.04
foreman	W05	72.5	0.11	2.39	0.05	1.33	85.63	0.36	5.25	-0.35	-2.01	80.45	-0.02	2.32	-0.05	0.31
	W075	73.99	-0.11	1.48	0.03	0.04	85.76	-0.41	5.69	-0.01	5.63	78.8	-0.05	2.57	0.05	2.63
	W09	76.12	-0.32	9.28	0.02	-2.17	89.8	-2.53	45.19	0.05	15.62	81.88	-0.03	2.25	-0.09	-3.2
	W01	83.76	-0.04	0.95	-0.22	0.67	90.82	0	2.62	-1.48	1.88	85.26	0	1.34	-0.04	0.15
	W025	83.74	-0.15	-0.87	-0.18	0.72	88.34	-0.11	1.32	0.48	2.79	82	-0.04	0.67	0.24	1.48
crew	W05	77.57	-0.19	5.9	-0.12	3.99	89.71	-0.13	6.44	-0.11	5.73	79.27	-0.07	1.08	0.03	1.81
	W075	81.19	-0.21	6.42	-0.12	3.34	92.44	-0.77	2.49	0.2	5.01	82.44	-0.07	0.68	0.05	2.44
	W09	81.77	-0.25	2.68	0	1.3	94.64	-0.68	2.09	0.1	1.81	89.29	-0.06	2.18	0.11	-0.51
Average		77.71	-0.08	1.92	-0.06	0.63	86.74	-0.22	4.62	0.02	2.83	82.36	-0.02	1.59	0.07	1.31

表 13 BLQP = 30 ; ELQP = 26 不同演算法的比較

Sequence	Weighting	Lin's					Zhao's					Proposed				
		BL		EL			BL		EL			BL		EL		
		TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR
	W01	83.5	-0.02	0.36	-0.14	0.97	86.98	-0.01	1.21	-0.95	5.09	80.04	0.02	1.18	-0.1	2.57
	W025	84.06	-0.07	-0.67	-0.12	1.04	87.09	-0.05	-0.4	0.55	-0.73	80.7	-0.05	0.33	0.21	0.29
mobile	W05	73.15	-0.06	-0.02	0.04	-0.29	85.06	-0.02	1.2	0	0.53	81.63	0.01	0.84	0.02	0.37
	W075	74.19	-0.25	3.02	0.06	0.55	87.43	-0.78	10.07	0.18	6.06	79.17	-0.02	2.04	0.06	1.17
	W09	75.98	-0.47	8.58	0.07	-1.67	89.45	-1.69	13.84	-0.12	1.6	78.89	-0.04	3.14	-0.17	-1.63
	W01	83.84	-0.04	0.34	-0.15	1.08	85.86	-0.12	0.84	1.41	2.13	78.16	-0.05	0.62	0.09	1.51
	W025	83.5	-0.09	-0.35	-0.07	1.22	85.97	-0.11	1.44	0.04	2.37	81.15	-0.1	1.06	-0.01	2.03
football	W05	70.26	-0.02	0.45	-0.02	0.45	85.19	0.11	3.33	-0.18	1.91	80.32	0.04	2.75	-0.11	1.86
	W075	70.45	0.18	1.54	-0.04	0.24	84.49	-0.12	2.78	-0.1	2.92	78.94	-0.06	1.75	-0.04	2.11
	W09	74.41	-0.1	1.92	-0.08	1.01	84.18	-0.92	0.17	-0.18	3.1	80.56	-0.07	0.75	-0.01	1.56
	W01	82.56	-0.01	0.29	-0.06	2.14	86.39	0	1.36	-0.02	4.07	84.01	0.01	1.27	-0.04	4.19
	W025	83.8	-0.08	-1.48	-0.13	0.86	86.39	-0.08	-0.47	0.38	1.05	84.43	0	0.44	0.13	1.27
harbour	W05	73.9	0.05	1.64	0.02	-0.04	81.78	0.05	2.19	-0.01	0.6	83.15	-0.07	0.11	0.02	0.62
	W075	75.25	-0.01	3.44	0.04	0.35	80.76	0	1.72	0.05	1.19	82.49	-0.05	4.01	0.07	2.35
	W09	75.79	-0.24	3.1	0.09	-1.31	84.01	-0.21	3.07	0.01	-2.21	82.75	-0.08	1.95	0.02	-0.53
	W01	83.6	-0.03	0.95	-0.25	2.27	88.65	-0.09	2.99	-0.97	9.78	82.05	0.02	2.64	-0.05	3.24
	W025	84.38	-0.07	0.49	-0.16	2.91	88.76	-0.1	1.7	0.1	3.51	83.17	-0.05	1.71	0.05	3.02
foreman	W05	73.15	-0.05	1.98	0.11	0.68	87.42	-0.15	3.77	0.02	6.06	81.22	-0.11	2.37	0	2.11
	W075	74.94	-0.2	2.13	0.05	0.11	87.53	-0.72	6.11	0.02	7.62	76.64	-0.08	-0.45	0.03	0.5
	W09	76.94	-0.23	9.35	0.06	0	91.45	-3.73	17.68	0.11	2.72	78.31	-0.03	3.02	-0.03	-1.55
	W01	84.14	-0.04	-0.04	-0.15	-0.19	94.26	0.05	4.42	-3.2	-13.15	85.5	0.03	0.66	-0.06	-0.75
	W025	85.25	-0.16	-0.95	-0.25	2.07	90.32	-0.11	1.68	0.38	3.14	82.25	-0.05	0.83	0.23	2.03
crew	W05	77.91	-0.16	5.22	-0.01	2.96	90.7	-0.39	6.77	0.06	8.73	79.71	-0.1	1.01	0.13	2.5
	W075	82.32	-0.31	6.18	-0.06	2.98	94.71	-1.35	2.07	0.22	3.8	83.69	-0.04	2.39	0.04	0.89
	W09	84.06	-0.26	-2.15	0.02	0.88	95.24	-1.48	-21.02	-0.07	2.8	87.95	-0.02	0.95	0	2.2
Average		78.85	-0.11	1.81	-0.05	0.85	87.6	-0.47	2.74	-0.09	2.59	81.48	-0.04	1.49	0.02	1.36

4.2.2 實驗二

此實驗是觀察本論文所提出的精進方法是否在 Zhao's *et al.*[4]所提的演算法中亦能有好的效能表現。實驗數據顯示，本論文所提的精進方法在所提演算法中的確能增加 RD 效能。其中在某些 case 下 Zhao's *et al.*[4]所提出演算法加入作者所提出的精進方法後，編碼時間節省反而比作者所提演算法少。因為在 Zhao's *et al.*[4]原本的演算法中，序列中所找的停止點可能不是最佳的，因此在精進程序上必須耗費更多的編碼時間，才能有更好的 RD 效能。

表 14 BLQP = 22 ; ELQP = 18 Zhao's 演算法中加入所提出的精進方法

Sequence	Weighting	Zhao's					Zhao's (Add Proposed Refine)					Proposed				
		BL		EL			BL		EL			BL		EL		
		TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR
	W01	85.44	0.08	1.3	0.38	2.1	81.52	0.23	1.07	-0.03	0.23	83	0.11	1.86	0.08	2.2
	W025	86	-0.03	0.42	0.01	0.26	83.65	-0.01	0.68	-0.02	0.62	83.05	0.11	2.02	0.26	2
mobile	W05	85.65	-0.05	0.43	0.34	2.09	81.39	0.15	1.11	0.01	0.87	82.91	0.02	0.87	0.19	1.38
	W075	85.44	0.02	0.39	0.06	1.31	81.63	0.05	0.91	0.01	0.47	82.31	0.08	1.54	0.04	1.34
	W09	86.9	-0.25	4.27	0.02	0.91	83.31	0.02	0.32	-0.24	3.43	80.35	-0.03	2.2	0.01	0.41
	W01	86.26	-0.12	0.97	0.77	1.66	80.78	0.69	1.43	-0.1	0.45	80.98	-0.06	0.83	0.63	1.8
	W025	86.24	-0.12	1.69	-0.02	2.01	82.44	-0.04	1.76	-0.11	1.23	81.41	-0.1	1.87	-0.01	2.24
football	W05	85.68	-0.06	2.27	-0.12	1.99	83	-0.13	2.08	-0.08	2.21	84.29	-0.05	2.27	-0.04	2.09
	W075	85.47	0.42	2.9	-0.15	1.55	85.03	-0.07	1.63	0.32	2.59	82.1	0.25	2.43	-0.05	1.54
	W09	84.68	-0.22	1.72	-0.09	2.52	81.43	-0.07	2.42	0.3	2.5	78.28	-0.07	1.24	-0.02	1.45
	W01	84.83	0.07	1.6	0.19	1.99	83.71	0.23	1.44	0.03	0.91	83.78	0.07	1.57	0.23	1.97
	W025	87.38	0.26	3.62	0.54	4.22	85.91	-0.01	-0.01	-0.05	0.33	84.94	0.04	1.51	0.16	1.48
harbour	W05	87.51	0.03	1.5	0.06	1.03	86.81	0.07	0.7	0.01	0.52	85.61	-0.04	0.82	0.06	0.89
	W075	87.28	0.05	0.48	-0.01	0.2	86.72	-0.03	0.65	0.03	0.77	83.77	-0.01	0.58	-0.02	0.55
	W09	83.39	0.04	0.46	-0.01	-0.46	81.58	0	0.49	0.06	3.42	81.29	0.1	1.95	0.02	0.28
	W01	85.31	-0.03	1.56	-0.1	3.49	79.26	-0.28	1.63	-0.01	1.26	79.22	-0.03	1.14	-0.04	1.55
	W025	85.99	-0.05	0.83	0.45	1.44	81.17	0.19	1.68	-0.03	0.95	81.82	-0.03	1.07	0.25	2.12
foreman	W05	85.52	0.39	4.83	-0.48	-2.75	79.91	-0.14	-0.3	0.11	2.32	82.37	0.16	3	-0.05	-0.59
	W075	84.27	-0.29	0.84	0.01	2.08	82.47	0.03	2.28	-0.05	1.37	80.26	-0.09	2.38	0.02	2.53
	W09	87.19	-0.89	13.41	0.05	5.49	84.27	0.01	1.82	-0.33	7.42	79.06	-0.04	2.15	0.04	0.07
	W01	88.78	-0.05	1.92	-0.13	3.43	83.28	-0.25	1.26	-0.03	1.06	79.69	-0.02	0.83	-0.03	1.47
	W025	88.44	-0.13	2.33	0.24	3.02	82.38	0.19	0.82	-0.08	0.48	80.6	-0.06	1.45	0.26	1.83
crew	W05	87.98	0.48	6.3	-0.58	-1.78	80.45	-0.18	-0.48	0.22	2.92	81.5	0.27	2.88	-0.04	1.08

	W075	88.66	-0.22	1.84	0.05	3.92	84.96	0.08	2.4	-0.2	2.34	81.12	-0.02	2.54	0.09	2.83
	W09	93.04	-0.7	7.03	0.08	0.52	87.54	0.1	-0.5	-0.38	5.61	84.67	-0.04	2.33	0.06	-0.31
Average		86.53	-0.05	2.6	0.06	1.69	82.98	0.03	1.09	-0.03	1.85	81.94	0.02	1.73	0.08	1.37

表 15 BLQP = 26 ; ELQP = 22 Zhao's 演算法中加入所提出的精進方法

Sequence	Weighting	Zhao's					Zhao's (Add Proposed Refine)					Proposed				
		BL		EL			BL		EL			BL		EL		
		TS	ΔPSNR	ΔBR	ΔPSNR	ΔBR	TS	ΔPSNR	ΔBR	ΔPSNR	ΔBR	TS	ΔPSNR	ΔBR	ΔPSNR	ΔBR
	W01	87.65	-0.04	0.69	-0.26	4.04	81.59	0.06	1.83	0	0.72	83.82	-0.02	0.78	-0.02	1.23
	W025	85.87	-0.04	0.21	0.33	-0.3	82.51	0.32	0.1	-0.06	0.37	83.06	-0.01	0.95	0.31	0.45
mobile	W05	84.62	0.13	2.93	-0.03	0.41	80.36	0.01	0.34	-0.03	1.49	79.75	0.05	1.88	0	0.44
	W075	86.98	-0.17	3.67	0.39	4.94	83.28	0.35	4.46	-0.14	3.45	79.32	-0.05	2.15	0.24	3.01
	W09	87.44	-0.93	14.97	0.29	4.22	84.42	0.24	2.68	-0.26	4.51	79.88	-0.07	2.12	0.17	1.23
	W01	86.03	-0.11	0.63	1.3	2.09	81.56	1.01	1.85	-0.07	0.27	82.62	-0.04	0.33	0.25	1.89
	W025	86.13	-0.09	1.52	0.01	2.23	81.32	0.02	2.05	-0.07	0.9	82.19	-0.06	1.83	0.06	2.71
football	W05	85.76	0.01	2.57	-0.14	1.99	82.29	-0.11	1.75	0.04	2.32	80.9	0.04	2.66	-0.1	2.26
	W075	85.43	0.78	4.17	-0.15	1.32	84.19	-0.06	1.6	0.33	3.06	80.06	0.05	1.67	-0.05	1.77
	W09	84.98	-0.36	1.04	-0.11	3.39	83.52	-0.05	1.99	-0.17	2.34	80.6	-0.01	0.83	0	0.54
	W01	84.64	-0.07	-0.14	-0.02	0.88	82.57	-0.12	1.11	-0.03	0.31	86.83	-0.03	0.83	-0.03	1.1
	W025	83.97	-0.07	-0.06	0.19	0.29	82.04	0.31	0.76	-0.05	0.22	83.8	-0.03	0.92	0.32	1.26
harbour	W05	82.49	-0.05	1	0.03	1.32	81.8	0.03	0.95	-0.05	0.66	81.66	0.06	1.89	0.04	1.29
	W075	82.51	-0.05	-0.22	0.02	0.43	80.53	0.03	0.99	-0.09	1.18	84.66	-0.04	1.71	0.03	1.32
	W09	83.26	-0.18	7.8	0	-1.67	82.01	0.08	0.84	-0.34	3.84	81.97	-0.04	2.03	0.07	0.13
	W01	87.47	0.04	3.65	-0.8	6.9	81.99	-0.49	2.64	-0.02	1.68	84.82	0.01	2.64	-0.03	3.86
	W025	86	-0.13	0.08	0.48	1.73	81.86	0.26	1.58	-0.06	0.25	83.56	-0.04	1.4	0.28	3.04
foreman	W05	85.63	0.36	5.25	-0.35	-2.01	81.03	-0.11	0.93	0.1	3.39	80.45	-0.02	2.32	-0.05	0.31
	W075	85.76	-0.41	5.69	-0.01	5.63	82.4	0	3.95	-0.14	4.25	78.8	-0.05	2.57	0.05	2.63
	W09	89.8	-2.53	45.19	0.05	15.62	85.65	-0.06	1.12	-0.9	12.56	81.88	-0.03	2.25	-0.09	-3.2
	W01	90.82	0	2.62	-1.48	1.88	84.77	-0.41	1.22	0	1.5	85.26	0	1.34	-0.04	0.15
	W025	88.34	-0.11	1.32	0.48	2.79	81.61	0.19	0.54	-0.12	-0.31	82	-0.04	0.67	0.24	1.48
crew	W05	89.71	-0.13	6.44	-0.11	5.73	84.28	0.04	2.06	0.01	1.72	79.27	-0.07	1.08	0.03	1.81
	W075	92.44	-0.77	2.49	0.2	5.01	89.95	0.15	2.45	-0.42	-0.07	82.44	-0.07	0.68	0.05	2.44
	W09	94.64	-0.68	2.09	0.1	1.81	91.26	0.13	0.4	-0.34	1.36	89.29	-0.06	2.18	0.11	-0.51
Average		86.74	-0.22	4.62	0.02	2.83	83.15	0.07	1.61	-0.11	2.08	82.36	-0.02	1.59	0.07	1.31

表 16 BLQP = 30 ; ELQP = 26 Zhao's 演算法中加入所提出的精進方法

Sequence	Weighting	Zhao's					Zhao's(Add Proposed Refine)					Proposed				
		BL		EL			BL		EL			BL		EL		
		TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR
	W01	86.98	-0.01	1.21	-0.95	5.09	83.28	-0.16	2.53	-0.01	1.34	80.04	0.02	1.18	-0.1	2.57
	W025	87.09	-0.05	-0.4	0.55	-0.73	81.82	0.18	0.23	-0.03	0.45	80.7	-0.05	0.33	0.21	0.29
mobile	W05	85.06	-0.02	1.2	0	0.53	80.56	0.04	0.74	-0.07	1.54	81.63	0.01	0.84	0.02	0.37
	W075	87.43	-0.78	10.07	0.18	6.06	84.29	0.1	3.33	-0.22	5.09	79.17	-0.02	2.04	0.06	1.17
	W09	89.45	-1.69	13.84	-0.12	1.6	86.53	-0.06	-0.41	-0.47	6.45	78.89	-0.04	3.14	-0.17	-1.63
	W01	85.86	-0.12	0.84	1.41	2.13	80.47	0.81	2.01	-0.05	1.08	78.16	-0.05	0.62	0.09	1.51
	W025	85.97	-0.11	1.44	0.04	2.37	80.92	0.04	2.19	-0.1	1	81.15	-0.1	1.06	-0.01	2.03
football	W05	85.19	0.11	3.33	-0.18	1.91	81.37	-0.13	1.87	0.08	2.65	80.32	0.04	2.75	-0.11	1.86
	W075	84.49	0.12	2.78	-0.1	2.92	82.92	-0.07	2.74	0.04	3	78.94	-0.06	1.75	-0.04	2.11
	W09	84.18	-0.92	0.17	-0.18	3.1	84.96	-0.01	0.95	-0.14	2.56	80.56	-0.07	0.75	-0.01	1.56
	W01	86.39	0	1.36	-0.02	4.07	85.83	-0.01	4.37	0.01	1.97	84.01	0.01	1.27	-0.04	4.19
	W025	86.39	-0.08	-0.47	0.38	1.05	84.93	0.07	0.69	0.02	0.21	84.43	0	0.44	0.13	1.27
harbour	W05	81.78	0.05	2.19	-0.01	0.6	80.14	0.03	0.71	-0.02	0.75	83.15	-0.07	0.11	0.02	0.62
	W075	80.76	0	1.72	0.05	1.19	78.05	0.05	3.02	-0.06	5.85	82.49	-0.05	4.01	0.07	2.35
	W09	84.01	-0.21	3.07	0.01	-2.21	81.79	0.13	-0.84	-0.38	1.69	82.75	-0.08	1.95	0.02	-0.53
	W01	88.65	-0.09	2.99	-0.97	9.78	83.03	-0.24	4.8	-0.01	1.96	82.05	0.02	2.64	-0.05	3.24
	W025	88.76	-0.1	1.7	0.1	3.51	83.15	0.01	1.4	-0.04	0.24	83.17	-0.05	1.71	0.05	3.02
foreman	W05	87.42	-0.15	3.77	0.02	6.06	81.91	0.02	3.87	-0.11	3.26	81.22	-0.11	2.37	0	2.11
	W075	87.53	-0.72	6.11	0.02	7.62	83.93	0.02	4.56	-0.24	5.7	76.64	-0.08	-0.45	0.03	0.5
	W09	91.45	-3.73	17.68	0.11	2.72	88.75	-0.1	-2.94	-0.87	8.2	78.31	-0.03	3.02	-0.03	-1.55
	W01	94.26	0.05	4.42	-3.2	-13.15	88.19	-0.31	-0.3	-0.03	1.63	85.5	0.03	0.66	-0.06	-0.75
	W025	90.32	-0.11	1.68	0.38	3.14	81.42	0.28	1.71	-0.05	1.45	82.25	-0.05	0.83	0.23	2.03
crew	W05	90.7	-0.39	6.77	0.06	8.73	83.24	0.08	3.37	-0.19	0.95	79.71	-0.1	1.01	0.13	2.5
	W075	94.71	-1.35	2.07	0.22	3.8	92.49	0.08	1.23	-0.43	0.25	83.69	-0.04	2.39	0.04	0.89
	W09	95.24	-1.48	-21.02	-0.07	2.8	93.82	0.03	1.52	-0.54	-6.17	87.95	-0.02	0.95	0	2.2
Average		87.6	-0.47	2.74	-0.09	2.59	83.91	0.04	1.73	-0.16	2.12	81.48	-0.04	1.49	0.02	1.36

4.2.3 實驗三

實驗三中要去觀察本論文所提供的快速編碼模式決策演算法，工作在不是用來統計資料的視訊序列中是否依然有相同的 RD 效能和編碼時間節省。實驗結果顯示本論文所提的快速演算法在其地的視訊序列中依然有不錯的 RD 編碼效能，且在編碼時間節省上也是將近有 82%。

表 17 其他不同的視訊序列 RD 效能

Sequence	Weighting	BLQP=22 ELQP=18						BLQP=26 ELQP=22						BLQP=30 ELQP=26					
		BL			EL			BL			EL			BL			EL		
		TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR	TS	Δ PSNR	Δ BR	Δ PSNR	Δ BR			
	W01	81.24	0.01	0.4	-0.07	0.08	81.24	0.01	-0.87	-0.02	-0.63	82.34	-0.01	-0.31	0.00	0.28			
	W025	81.34	-0.01	-0.12	0.07	-0.33	82.29	0.00	-0.09	-0.02	-0.88	83.15	-0.01	-0.40	0.06	0.98			
akiyo	W05	82.61	-0.05	-0.34	-0.01	-1.17	83.11	-0.02	-0.48	-0.01	-1.17	84.25	-0.07	-0.42	0.01	-0.14			
	W075	80.78	-0.07	-1.17	0.06	1.57	79.25	0.00	1.30	-0.02	-0.07	81.72	-0.05	0.83	0.01	0.34			
	W09	79.99	-0.06	0.7	0.06	0.15	80.14	-0.08	-0.05	0.00	0.46	80.3	-0.03	-1.04	0.00	0.15			
	W01	82.15	0.00	0.04	-0.08	0.34	82.9	-0.01	0.22	-0.04	0.81	82.81	-0.01	-0.03	-0.09	1.09			
	W025	83.41	0.00	0.34	0.07	0.87	83.11	-0.01	0.58	-0.01	0.74	84.55	0.01	0.83	0.03	1.20			
bus	W05	81.39	-0.02	0.63	0.00	0.69	81.63	-0.01	0.99	0.03	1.31	82.43	0.02	1.45	0.05	0.95			
	W075	78.64	0.01	1.12	-0.01	0.88	80.62	-0.06	1.64	0.01	1.99	81.6	-0.06	2.47	-0.01	2.72			
	W09	77.51	-0.02	1.45	-0.01	0.12	79.2	-0.04	1.94	0.03	-0.02	80.84	0.01	1.09	-0.02	-1.10			
	W01	84.01	0.02	0.49	-0.04	0.36	83.63	0.00	0.62	-0.12	1.19	84.42	0.04	0.22	0.12	0.79			
	W025	82.72	-0.03	0.86	0.01	0.76	81.67	0.01	0.97	0.03	0.70	82.81	0.03	0.42	-0.05	-0.09			
ice	W05	82.27	-0.02	1.39	-0.02	1.04	82.41	-0.05	1.19	-0.02	1.35	83.54	-0.01	0.78	-0.03	0.66			
	W075	81.51	-0.07	1.67	-0.02	1.31	80.91	0.04	0.68	0.00	0.66	82.37	-0.02	1.17	-0.01	0.84			
	W09	80.69	-0.05	1.91	-0.03	0.31	81.5	-0.06	0.93	-0.01	-0.52	81.05	-0.04	3.38	-0.03	-1.95			
	W01	82.05	0.00	0.19	-0.07	-0.10	83.52	-0.01	0.57	-0.03	0.41	83.42	-0.01	0.49	-0.02	-0.21			
	W025	83.17	-0.01	0.11	-0.04	0.09	82.33	0.00	0.45	-0.05	-1.02	83.52	-0.02	0.29	-0.01	-0.09			
silent	W05	81.22	-0.08	-0.51	0.02	0.16	82.14	-0.09	-0.78	0.00	-0.16	84.26	-0.02	0.73	0.02	0.96			
	W075	76.64	-0.08	0.60	-0.01	0.54	79.61	-0.01	0.18	0.02	0.81	78.62	-0.05	1.84	0.00	1.58			
	W09	78.31	-0.06	-0.41	0.02	0.33	80.25	-0.05	-0.36	0.01	0.17	79.24	-0.06	-0.10	0.00	0.45			
	W01	79.62	0.00	-0.02	0.20	0.80	80.05	0.00	0.01	-0.04	0.37	81.35	0.00	0.50	-0.04	1.69			
	W025	82.64	0.00	0.78	-0.05	0.79	81.61	-0.02	0.20	0.13	1.13	83.40	0.00	1.16	0.01	1.43			
soccer	W05	81.71	-0.08	0.22	-0.03	0.69	82.51	0.05	0.86	0.04	1.51	81.68	-0.06	1.38	0.00	1.27			
	W075	79.64	-0.03	1.01	0.03	1.46	80.61	-0.05	1.05	0.02	1.41	81.43	-0.03	2.20	0.00	1.32			
	W09	80.04	0.16	1.81	0.00	1.19	78.98	-0.06	2.31	-0.01	-0.07	80.68	-0.04	1.53	-0.03	-0.42			
Average		81.01	-0.02	0.53	0.00	0.52	81.41	-0.03	0.56	0.00	0.42	82.23	-0.02	0.82	0.00	0.59			

第五章 結論與未來展望

本論文提供了一個快速編碼模式決策演算法，其主要是改善 Zhao's *et al.*[4] 所提演算法中只考慮編碼模式序列中的停止位置到下一個更好的編碼模式位置之間的距離有多遠，但這個距離並不能給予任何保證。基於這個原因，本論文提出多加入考慮 RD-cost 下降量和不同編碼模式配對之間的運算複雜度。利用以上兩點去找出序列中最佳的停止位置。此外本論文另外提出一套精進的動作，能讓快速演算法的 RD 效能更好，並且此編碼模式決策的精進程序也能適應在不同的演算法上。最後實驗結果顯示本論文所提出的快速演算法在 RD 效能方面和完整搜尋法相比，其平均的位元率和峰值信噪比幾乎相同。並且比 Zhao's *et al.*[4] 所提的演算法和 Lin's *et al.*[6] 所提的演算法有更穩定的 RD 效能。並且本論文所提快速編碼模式決策演算法在編碼時間節省上平均有 82% 左右。

本論文所提演算法只實做在可調視訊編碼的 CGS 架構且畫面之間的架構是 IPPP 架構，也就是只工作在 I、P 畫面上，在未來可以多加入 B 畫面的考慮，讓所提演算法能適應更多不同的設定架構上。然而在我所考慮的不停編碼模式配對的運算複雜度是利用整個視訊序列的編碼時間，但此時間可能不是真正的編碼模式配對的運算時間，在未來若能更詳細的分析並能更準確的計算出各個不同編碼模式配對的運算時間，想必在編碼模式配對序列中所找到的停止點能更加準確，演算法必能有更好的效能提升

參考文獻

- [1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103-1120, 2007.
- [2] T.S. Ferguson, Optimal Stopping and Applications, [Online] Available: <http://www.math.ucla.edu/~tom/Stopping/Contents.html>.
- [3] T.S. Ferguson, J.P. Hardwick and M. Tamaki, "Maximizing the Duration of Owning a Relatively Best Object," in "Strategies for Sequential Search and Selection in Real Time", *Contemp. Math.*, vol. 125, pp. 37-57, 1992.
- [4] T. Zhao, S. Kwong, H. Wang and C.-C. J. Kuo, "H.264/SVC Mode Decision Based on Optimal Stopping Theory and All-Zero Block Detection,"
- [5] H. Schwarz and T. Wiegand, "R-D Optimized Multi-layer Encoder Control for SVC," *IEEE Int'l Conf. on Image Processing*, 2007.
- [6] J.Y. Lin and W. H. Peng, "A Fast Mode Decision Algorithm for SVC Multi-Layer Encoder Control," 2009.
- [7] T. France, H. Schwarz, and M. Wien "JSVM Software (version 9.12.2)," *Joint Video Team*, JVT08/Z203, 2008.
- [8] S. T. Kim, K. R. Konda and C. S. Cho, "Fast Mode Decision Algorithm for Spatial and SNR Scalable Video Coding," *IEEE International Symposium on Circuits and Systems (ISCAS 2009)*, May 2009, pp. 872-875.
- [9] L. Shen, Z. Liu, P. An, R. Ma, Z. Zhang, "An adaptive early termination of mode decision using inter-layer correlation in scalable video coding", *Proc. IEEE International Conference on Image Processing*, ICIP-2010, Hong Kong, September 2010.
- [10] S.T. Kim, K.R. Konda, P.S. Mah, S.J. Ko. "Adaptive Mode Decision Algorithm for Inter Layer Coding in Scalable Video Coding", *Proc. IEEE International Conference on Image Processing*, ICIP-2010, Hong Kong, September 2010.
- [11] A. Huang, X. Lin, Y. Chen, "Fast Mode Decision Algorithm for Spatial and Coarse Grain Quality Scalable Video Coding", *IEEE International Symposium on Computer Network and Multimedia Technology(CNMT 2009)*, vol. 1 and 2, pp. 199-202, 2009.
- [12] J. F. Ren and N. Kehtarnavaz, "Fast Adaptive Early Termination for Mode Selection in H.264 Scalable Video Coding," *Journal of Real-Time Image Processing*, v. 4, n. 1, pp. 13-21, Mar. 2009.

- [13] J. F. Ren and N. Kehtarnavaz, "Fast Adaptive Early Termination for Mode Selection in H.264 Scalable Video Coding," in *Proc. IEEE ICIP'08*, pp.2464-2467, Oct. 2008

