

H.264 視訊編碼中之快速場景轉換偵測及快速模式決定演算法

曾伶玉

國立中興大學資訊網路與多媒體研究所

國立中興大學資訊科學與工程學系

Email: lytseng@cs.nchu.edu.tw

楊詠竣

國立中興大學資訊網路與多媒體研究所

Email: kazuya280@yahoo.com.tw

麥恩嘉

國立中興大學資訊網路與多媒體研究所

Email: enjia.mai@msa.hinet.net

摘要—新一代的視訊壓縮標準 H.264/AVC 較前一標準 MPEG4 大約增加了 1.5 倍的壓縮效能，但是卻在計算複雜度上增加了超過十倍的昂貴代價。其中計算量最大的地方就是在模式決定(Mode decision)以及其中移動估計(Motion Estimation)的部分。本論文提出視訊場景轉換偵測演算法與模式決定的快速演算法，藉由觀察亮度直方圖的差(Histogram Difference)與設定門檻值的方式，在場景轉換的情況可減少 H.264 不必要的移動估計時間，另一方面，在模式決定過程中僅增加些微的位元率，卻可比 H.264 減少約 35% 的編碼時間。

關鍵詞—H.264/AVC、場景轉換偵測、模式決定、移動估計

一、緒論

隨著多媒體系統與通訊傳輸應用的不斷推陳出新，視訊影像壓縮對於視訊資料的傳輸與儲存，扮演極重要的角色。H.264/AVC 是由 JVT 在 2003 年制定的最新視訊壓縮標準[7]。H.264 標準是現行標準中壓縮率最高的，與先前的標準相比，在同樣的畫面品質下，它可省下一半左右的位元率，原因是它導入許多先前標準所沒有的技術，如：可變區塊大小、四分之一像素精確度搜尋，多重參考畫等。由於進行了分層設計，H.264 也具有更好的容錯性。H.264 雖然能有較好的壓縮率與影像品質，但卻帶來龐大的計算複雜度，造成即時編碼系統實現的困難度。

H.264 和先前的編碼標準一樣採巨區塊(macro block)的編碼方式，也就是以 16x16 大小的巨區塊做為編碼的基本單位，每個巨區塊又分成兩種編碼方式：畫面間編碼(inter-coding)和畫面內編碼(intra-coding)。H.264 的模式決定(mode decision)即是要找出最佳的編碼模式，對每一個 P-畫面的巨區塊來說，必須從畫面內及畫面間兩種編碼方式的各個不同模式中，選擇出最好的編碼方式。H.264 使用位元率-失真最佳化(rate-distortion optimization)來計算編碼成本，使得編碼出來的品質(PSNR)能維持一定水準的情況下位元率(bit rate)亦可最小。區塊切割模式的決定在 H.264 視訊壓縮中扮演重要的角色，它會影響壓縮的品質與速率，而原本完全模式的決策演算法，雖可找出最適合的區塊切割，但耗費時間過於龐大。

回顧近幾年在各國際期刊中提出之快速模式決定演算法，像 Yin 等人提出了一種快速模式決定演算法[9]，主要概念就是藉由 RD cost 的遞增性，加上與預先設定的門檻值做比較，選擇性的跳過某一些模式的計算。而 Wu 等人所提出的方法[8]，是透過 Sobel 邊緣運算子所得到的資訊來判斷是否符合 spatial homogeneous 以及 temporal stationary，如果是就會被以 inter16x16 或 skip mode 來編碼，並省略掉其他模式的計算而達到加速的效果。接著 PAN 等人

[5] 提出同樣用 Sobel 邊緣運算子來求得邊緣斜率相似的候選模式，且只計算 DC 模式和候選的模式之 RD cost，以減少計算的時間。Choi 等人提出一個快速演算法[1]，藉由計算平均邊界像素錯誤值(average boundary error, ABE)來檢測巨區塊與相鄰巨區塊的空間相關性大小，以及計算移動補償的殘差資料之平均編碼位元率(average rate, AR)檢測時間相關性的高低。利用這兩個資訊來判斷是否要作 intra 模式的計算。Yu 等人提出三個階段的快速模式決定演算法[10]。第一個階段先用 AZB(all zero block) 偵測演算法決定是否要直接設定為 skip mode，第二階段檢查其 AC 係數的變異數觀察其平滑程度，進而決定是否進行 SUB 8x8 模式的計算，而第三階段就進行 SUB 8x8 的 RD cost 計算。Liu 等人提出的方法[4] 是先把所有的 inter 模式分成五個子集，然後憑藉著在正規化的 motion vector field 中計算 motion homogeneity，進而決定每個巨區塊中候選的 inter-mode 的子集，非此子集中的模式就略過不計算。近年 Kim 等人提出的快速模式決定演算法[2]，另外紀錄一個 AMVM(adaptive motion vector map)，等於參考前一個畫面與前前一個畫面之間是否有發生過 intra 模式，以及計算巨區塊四周鄰居區塊與自己的移動程度 NASMV(neighborhood absolute sum of motion vector) 及 ASMV，判斷是否進行 intra 模式的計算。Lee 與 Shin 所提出的方法[3]，是以 HD 值(histogram difference)及 Rate 為模糊邏輯(fuzzy logic)的 membership function，利用模糊邏輯進行產生的權重值來衡量是否進行 intra 模式的計算，以減少計算時間。

本論文的架構說明如下：第二節介紹我們所提出的視訊場景轉換偵測演算法。第三節介紹我們的快速模式決定演算法。第四節分別列示出兩個演算法的實驗結果。最後一節是我們的結論與

未來研究方向。

二、視訊場景轉換偵測演算法

在本節中將介紹本文所提出的場景轉換偵測演算法之設計準則和流程，H.264 編碼流程裡有畫面內與畫面間預測，在編碼一個巨區塊時，編碼器會先進行畫面間預測，計算 inter16x16、16x8、8x16、8x8 區塊的 RD-cost，接著計算子區塊分割裡 inter8x8、8x4、4x8、4x4 的 RD-cost，最後求出畫面內模式 intra4x4 跟 intra16x16 的 RD-cost，RD-cost 最小者為該巨區塊最佳編碼模式。H.264 是以最終編碼成本當作畫面間與畫面內預測的判斷依據，當畫面為平滑移動時，巨區塊基本上不需畫面內預測，該部分所花費的計算即成為一種浪費；當畫面有新的物件進入時，該新物件區塊無法從參考畫面找出合適的移動向量，最終結果會屬於畫面內預測，在此一情況，畫面間預測的計算也成為浪費。

編碼流程將畫面內預測排在最後計算其原因是，對於一段視訊序列，一張畫面裡會判給畫面內預測的出現機率太低，由表 1 可發現，在某些測試影片中，畫面內預測出現的機率甚至不到 1%。

表 1 Intra prediction 所佔比率

video sequence	SKIP mode		inter16x16,16x8,8x16,8x8		inter8x8,8x4,4x8,4x4		Intra prediction	
	MB	%	MB	%	MB	%	MB	%
Miss	92	92.93	7	7.07	0	0	0	0
Suzie	65	65.66	30	30.30	4	4.04	0	0
Akiyo	97	98.96	1	1.01	1	1.01	0	0
Forman	27	27.27	42	42.42	25	25.25	5	5.05
Silent	68	68.68	13	13.13	13	13.13	5	5.05

但是，當視訊序列中發生場景轉換的情形時，整張畫面與前一張畫面完全不相像，可以明顯發現，整張畫面內所有巨區塊做的畫面間預測大都是浪費。此種情況最容易發生在有大量畫面切換的影片，如：球賽轉播等。

為了解決此一問題，在編碼之前先判斷上述情況發生與否，若發生場景轉換，則跳過畫面間預測編碼，如此一來即可減少不必要的計算。然而，因每次編碼前都需要事先判斷，為了不在場景無轉換時增加編碼負擔，演算法勢必得簡化。

對於兩張完全不相像的影像，人眼可立刻判別出影像不同，即便是兩張很小的影像也可以判別出來。依此一特性，可以得知判斷時並不須採用原本影像大小。以兩張完全不相似的影像做差之絕對值和(Sum of Absolute Difference, SAD)運算，觀察其數值可知 SAD 越大畫面不相像的可能性相對提高。然而，在視訊序列中常發生物件從畫面外進入或鏡頭跟著前景移動，因此單以 SAD 運算用在視訊序列上對判別場景轉換仍容易出錯。一張影像的亮度直方統計圖，代表著該影像的亮度分布圖，相像的影像亮度直方圖分佈一定類似，如圖 1，所以我們利用亮度直方圖與 SAD 的原理，設計出場景轉換之偵測演算法。流程如圖 2 所示。

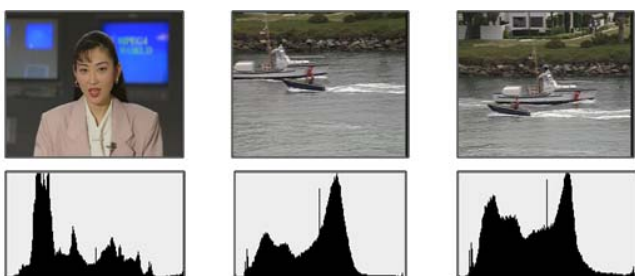


圖 1 各畫面亮度直方圖

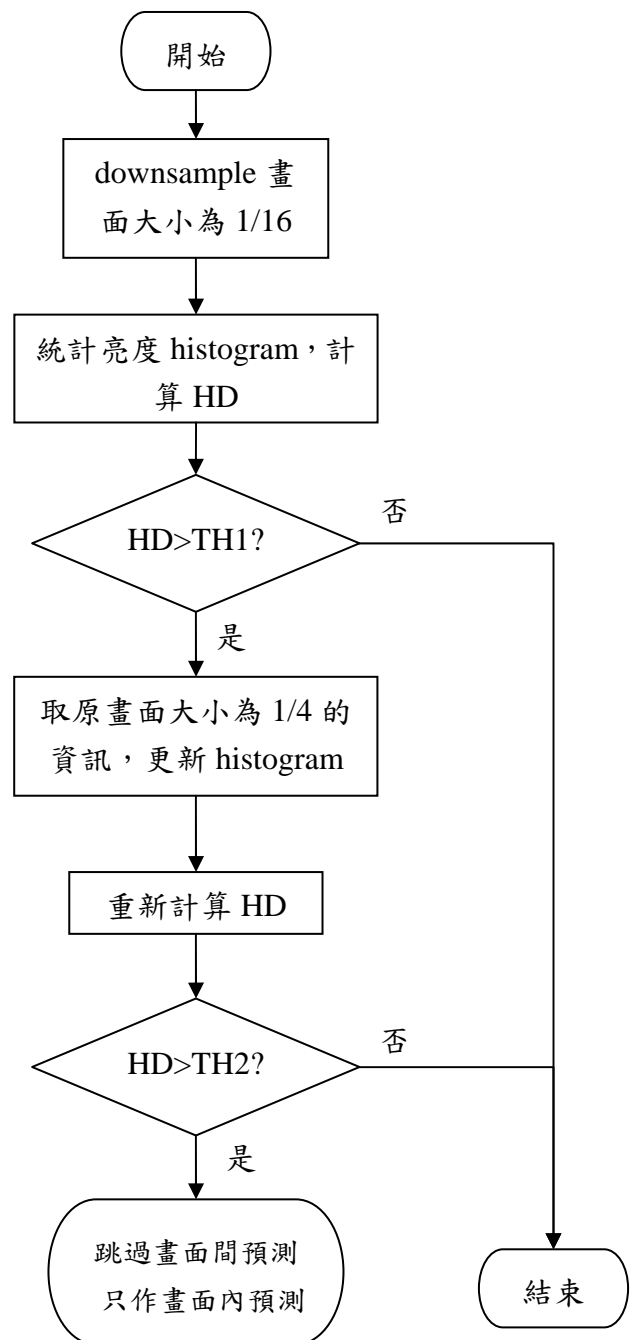


圖 2 場景轉換偵測演算法流程圖

整個偵測過程為：因畫面切換發生在整張畫面，所以需以畫面為單位進行處理。對於一張新進入的畫面，先把亮度分量畫面的長寬各 downsample 為 1/4，即面積縮小為 1/16，將縮小

過的畫面作亮度直方圖統計。當下一張畫面進入時統計 1/16 畫面的亮度直方圖，將該張直方圖與前一張直方圖做 SAD 運算，此一運算稱計算直方圖的差 (Histogram Difference, HD)。當 1/16 畫面的 HD 大於門檻值 TH1 時，再將兩張畫面 downsample 為 1/4 大小之畫面資訊統計到各自的亮度直方圖中，重新計算 HD，如果 1/4 畫面之 HD 小於門檻值 TH2，則演算法結束，該畫面內所有巨區塊的編碼過程無須調整；當 HD 大於門檻值 TH2 時，判斷該畫面為場景轉換，傳送跳過畫面間預測的訊號給新進畫面內所有的巨區塊。當巨區塊收到訊號，編碼時即會跳過畫面間預測。

三、快速模式決定演算法

本節介紹本論文所提出的快速模式決定演算法。模式決定(mode decision)就是從畫面內及畫面間兩種編碼方式的各個不同模式中，要找出最佳的編碼模式，H.264 使用位元率-失真最佳化(rate-distortion optimization)來計算編碼成本，也就是以 RD cost 值最小的模式為最佳模式。但是 RD cost 的計算又使得計算複雜度變高，我們的快速演算法便是設法盡量減少某些模式的 RD cost 的計算，達到加速的效果。

(一) Early skip

H.264 中在 inter 模式中有一種 skip mode(又稱 copy mode), skip mode 必須滿足以下條件：

- 1.最佳的模式必須是 inter 16x16
- 2.最佳的移動向量必須和預測的移動向量相同
- 3.量化後的係數必須全部為零

因為在 skip 模式，編碼端只會傳送一個 flag 表示目前的巨區塊是此模式，也不需要傳送 residual 到解碼端，因此在位元率上是非常的少，也就是說在一個 video sequence 中，如果 skip 模式佔的比率越高，位元率一定會大量下降。另外，

如果能夠提早判斷是不是 skip 模式，也可以大量省下其他所有模式的計算時間。但是 skip 模式不能夠誤判太多，因為 skip 模式並不會傳送 residual，如果把 residual 比較大的巨區塊誤判為 skip 模式，雖然位元率會降低，速度也會比較快，但是會導致視訊品質(PSNR)的降低。

在這裡使用一個簡單的判斷是否要進行 early skip 的方法，就是先計算出 skip 模式的 cost，把 cost 跟預先設定的一個門檻值 TH_skip 作比較，如果 cost 小於這個門檻值就直接選擇最佳模式為 skip 模式。

(二) 設定門檻值以省略計算

1. 快速決定 inter 16x16 模式：

當已經進行過 early skip 的判斷步驟之後，我們在這裡假設如果目前巨區塊的 inter 16x16 的 RD cost 很小時，其以 inter 16x16 來編碼的機率是相當的高。所以當計算出的 inter 16x16 的 RD cost 後，就與一個預先設定的門檻值 TH_1 作比較，如果 RD cost 小於這個門檻值就直接選擇最佳模式為 inter 16x16 模式。

2. 略過 P8x8 模式：

藉由觀察 inter 16x16、inter 16x8 以及 inter 8x16 的 RD cost 的趨勢，來決定要不要略過 P8x8 的計算，當已經計算過 inter 16x16、inter 16x8 以及 inter 8x16 的 RD cost 之後，比較這三種模式的 RD cost，如果 inter 16x16 的 RD cost 最小，那就直接將最佳模式設為 inter 16x16，並且直接省略掉 P8x8 的計算。如果 inter 16x16 的 RD cost 不是最小的，那麼仍需繼續往下計算 P8x8 的模式。

3. 略過 Intra 模式：

根據之前文獻的研究結果[2] [3] 發現，intra 模式在一個 video sequence 中所佔的比率非常低，但是其所佔的計算量卻相當的大。此外我們在實

驗中也發現，在巨區塊中即使找不到正確的移動向量，但是這樣重建出來的 inter 預測畫面也常常比 intra 模式的預測畫面還好。因此我們的演算法傾向於儘量不去計算 intra 模式。因此當計算出 P8x8 的 RD cost 之後，如果 P8x8 的 RD cost 如果太大，大概可以推測用 inter 模式編碼可能無法得到很好的效果，所以這裡再設定一個門檻值 TH_2，如果 P8x8 的 RD cost 大於這個門檻值，那麼就直接用 intra 4x4 模式來編碼。

(三) 演算法流程

整個快速模式決定演算法的步驟如下：

Step 1: 計算 skip 模式的 cost 並與 Th_skip 比較。如果 skip 的 cost 小於 Th_skip，直接決定 skip 模式為最佳模式。否則繼續進行 step 2。

Step 2: 計算 inter 16x16 的 RD cost 並與 TH_1 比較。如果 inter 16x16 的 cost 小於 TH_1，直接決定 inter 16x16 為最佳模式。否則繼續進行 step 3。

Step 3: 計算 inter 16x8 和 inter 8x16 的 RD cost。比較 inter 16x16、inter 16x8 和 inter 8x16 的 RD cost，如果 inter 16x16 的 RD cost 最小，直接決定 inter 16x16 為最佳模式。否則繼續進行 step 4。

Step 4: 計算 P8x8 的 RD cost。比較 P8x8、inter 16x8 和 inter 8x16 的 RD cost，如果 inter 16x8 和 inter 8x16 其中一個最小，直接決定該模式為最佳模式。反之，如果 P8x8 最小，繼續進行 step 5。

Step 5: 比較 P8x8 與 TH_2。如果 P8x8 的 RD cost 大於 TH_2，直接決定最佳模式為 intra 4x4。否則決定最佳模式為 P8x8。

圖 3 為快速模式決定演算法的流程圖。

(四) 動態調整門檻值

使用設定門檻值的方法有一個問題存在，不同的視訊序列(video sequence)所適用的門檻值

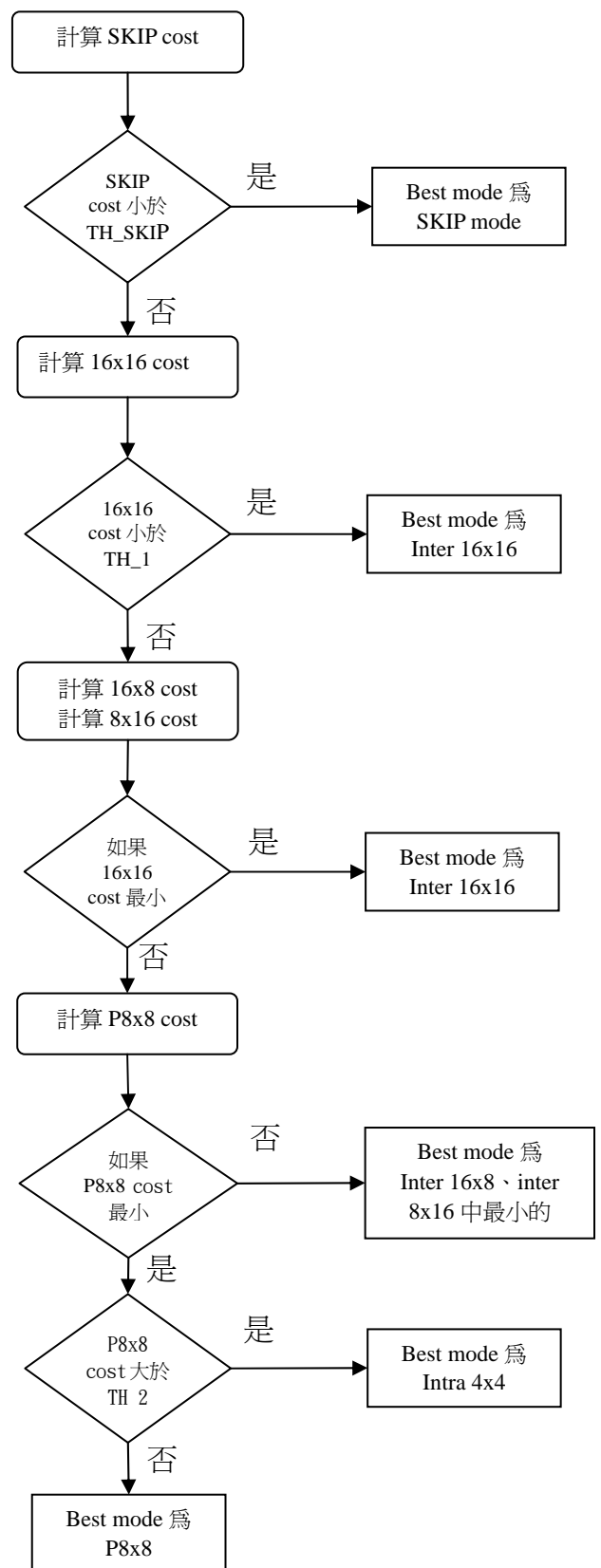


圖 3 快速模式決定演算法流程

並不相同，也就是說，同一組的門檻值並不適用於所有的視訊序列，有鑑於此，我們提出一個可以動態調整門檻值的方法，使用目前畫面與前一畫面的亮度直方圖的差(histogram difference)的量測來動態設定門檻值：

histogram difference(HD)的計算方式定義如下：

$$HD = \sum_{n=1}^{N_f} |h_i(n) - h_{i-1}(n)| \quad (1)$$

h_i 及 h_{i-1} 分別表示目前畫面與前一畫面的 histogram, N_f 表示整張畫面大小的像素數的四分之一。

動態調整門檻值先將目前畫面與前一張畫面 down sample 至四分之一，down sample 至四分之一的原因是可以減少計算量，卻又能夠保有資料的代表性。然後計算其 HD 值，這個 HD 值代表的意義是目前畫面與前一畫面的全域變化量的大小，HD 值越大，表示整個畫面變化的越大，也就是說，編碼方式會比較傾向於 intra 模式。反之 HD 值越小，代表整個畫面的變化並不大，編碼方式會傾向於使用 inter 模式。

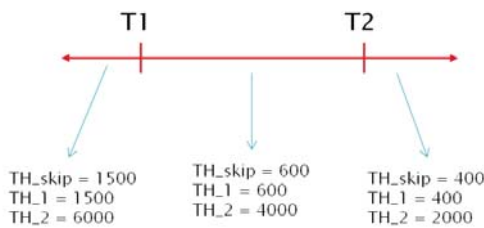


圖 4 動態門檻值的設定

因此我們以這個 HD 值的大小作為一個衡量基準，動態去調整每一種門檻值的方法，當計算出來的 HD 值座落於不同的範圍時，就使用不同組的門檻值來進行判斷，以圖 4 來說，當 HD 值小於 T1 就使用第一組門檻值，以此類推。這些

門檻值的設定所代表的意義為將採用的編碼方式傾向 intra 模式或是 inter 模式。

四、實驗結果

實驗的結果分為兩個部分，第一部分為第二節提出的視訊場景轉換偵測演算法，第二部分為第三節所提出的以快速模式決定演算法進行 H.264 的編碼，並且與其他人所提出的快速演算法作比較。實驗平台為個人電腦，處理器為 Intel Core 2 Duo T5500(1.66G)，記憶體為 1G，我們所提出的方法是以 JM 14.2 版進行修改，實驗的參數如下表格 2 所示：

表格 2 JM 參數設定

Parameters	Conditions
Quantization Parameters(QP)	28
Profile	Baseline
Search Range	16
Entropy coding	CAVLC
Sequence Type	IPPP
Reference Frame	1
RD Optimization	High complexity mode
Fast Chroma Decision	On
Sub pixel	1/4 pixel

實驗數據中 Bit rate 的計算方法為：

$$\Delta \text{Bitrate} = \frac{\text{Bitrate}_p - \text{Bitrate}_{JM}}{\text{Bitrate}_{JM}} \times 100\% \quad (2)$$

上述式子中 Bitrate_p 與為本快速演算法的位元率, Bitrate_{JM} 為 JM14.2 參考軟體的 bit rate, $\Delta \text{Bitrate}$ 為位元率的變化。

PSNR 的計算方法為：

$$\Delta \text{PSNR} = \text{PSNR}_p - \text{PSNR}_{JM} \quad (3)$$

上述式子中 $PSNR_p$ 中與為本快速演算法的 $PSNR$, $PSNR_{JM}$ 為 JM14.2 參考軟體的 bit rate, $\Delta PSNR$ 為 $PSNR$ 的變化。
時間的計算方法為：

$$\Delta T = \frac{T_p - T_{JM}}{T_{JM}} \times 100\% \quad (4)$$

上述式子中 T_p 與為本快速演算法的計算時間, T_{JM} 為 JM14.2 參考軟體的計算時間, ΔT 為時間的變化。

QCIF 格式的大小為 176×144 , CIF 格式大小則為 352×288 , 在實驗中的每個視訊序列 (video sequence) 的特性簡述如表格 3：

表 3 視訊序列的特性

Video sequence	特性
Foreman	人物有著中等程度的位移, 背景有移動。
News	物件有著中等程度的位移以及較少的細節, 背景幾乎不動。
Silent	人物有著中等程度的位移以及較少的細節, 背景幾乎不動。
Mobile	背景比較複雜, 背景與物件都有中等的位移。
Pairs	較複雜的背景, 人物的有中等的位移。
Stefan	複雜的背景, 人物有快速位移, 背景也會移動。
Bus	複雜的背景, 物件有快速位移, 背景也會移動。
Tennis	人物的有中等的位移, 場景切換兩次。

(一) 場景轉換之偵測演算法

場景轉換之偵測演算法實驗採用 3 個測試影片, 前兩個分別是 Bus 與 Tennis, 第三個影片為我們自行製作, 因目前研究用來比較的測試影片

並無加入轉場特效的影片, 為了觀察我們所提方法是否適合此情況, 挑選 4 段測試影片, 剪接後合併, 並在中間加入溶解 (cross dissolve)、滲入 (non-additive dissolve)、擺入 (swing in) 三種轉場特效, 長度為 24 張畫面, 格式為 YUV 4:2:0, 大小為 352×288 。

表 4 場景轉換偵測演算法測試結果 (bus) 搜尋範圍 16

Bus	Bit rate(kb/s)	Time(s)	PSNR(dB)
H.264 JM	1538.19	340.566	35.890
Our method	1538.19	340.589	35.890

表 5 場景轉換偵測演算法測試結果 (Bus) 搜尋範圍 32

Bus	Bit rate(kb/s)	Time(s)	PSNR(dB)
H.264 JM	1538.19	780.860	35.894
Our method	1538.19	781.316	35.894

表 6 場景轉換偵測演算法測試結果 (Tennis) 搜尋範圍 16

Tennis	Bit rate(kb/s)	Time(s)	PSNR(dB)
H.264 JM	979.51	213.804	34.751
Our method	976.50	210.919	34.751

表 7 場景轉換偵測演算法測試結果 (Tennis) 搜尋範圍 32

tennis	Bit rate(kb/s)	Time(s)	PSNR(dB)
H.264 JM	969.15	489.094	34.890
Our method	967.69	481.340	34.896

表 8 場景轉換偵測演算法測試結果(自製影片)

搜尋範圍 16

	Bit rate(kb/s)	Time(s)	PSNR(dB)
H.264 JM	1861.63	53.986	37.517
Our method	1889.35	47.736	37.518

表 9 場景轉換偵測演算法測試結果(自製影片)

搜尋範圍 32

	Bit rate(kb/s)	Time(s)	PSNR(dB)
H.264 JM	1878.42	123.052	37.513
Our method	1894.66	104.584	37.519

表 10 場景轉換畫面之比較 (Tennis)

tennis	frame no.	bit/frame	PSNR(dB)	inter mode	intra mode
H.264	67	104872	34.97	322	8
Our method	67	103776	34.94	0	330
H.264	97	125128	34.67	244	86
Our method	97	126592	34.91	0	330

由實驗結果可知，我們提出的場景轉換之偵測演算法在碰到場景轉換時，該張畫面可比原本的 JM 軟體少掉了做移動估計的時間，當移動向量搜尋範圍越大節省時間越多，且 PSNR 與 JM 軟體十分相近；當整個影片序列都沒有場景轉換情況下，總計算時間則比 JM 軟體計算時間多出不到 1% 時間，PSNR 則完全與 JM 軟體相同。由此可知本論文所提出的方法可在不增加計算負擔、不影響編碼畫面品質下，有效偵測出場景轉換。對於有轉場特效的部分，雖然稍微犧牲了位元數，但對於計算時間及 PSNR 也都有達到改善的作用。

(二)快速模式決定演算法

表 11 和表 12 分別列出在 QCIF 和 CIF 下，我們所提出的快速演算法與其他論文比較之結果。

表 11 QCIF 實驗結果比較

sequence		Our method	Lee[3]	Choi[1]	Wu[8]
Foreman QCIF	△Bit-rate(%)	1.84	0.35	0.43	1.28
	△PSNR	-0.012	0	-0.01	-0.062
	△Time(%)	-31.95	-28.09	-17.34	-25.18
News QCIF	△Bit-rate(%)	1.86	0.30	0.35	1.18
	△PSNR	0	-0.002	-0.01	-0.065
	△Time(%)	-36.52	-25.56	-27.15	-42.65
Silent QCIF	△Bit-rate(%)	2.14	0.23	-0.01	0.47
	△PSNR	0.037	-0.01	0.01	-0.022
	△Time(%)	-34.78	-25.56	-25.14	-45.16

表 12 CIF 實驗結果比較

sequence		Our method	Wu[8]
Mobile CIF	△Bit-rate(%)	0.34	0.13
	△PSNR	-0.05	-0.005
	△Time(%)	-35.32	-9.97
Paris CIF	△Bit-rate(%)	0.97	0.87
	△PSNR	-0.012	-0.040
	△Time(%)	-36.43	-31.90
Stefan CIF	△Bit-rate(%)	0.81	0.33
	△PSNR	-0.041	-0.015
	△Time(%)	-34.80	-17.37

由實驗結果可以得知，我們所提出的演算法在品質(PSNR)上維持的不錯，位元率(bit-rate)比起其他論文所提出方法稍微差了一點，而在時間上比 Lee[3] 與 Choi[1] 在 QCIF 有著顯著的改進，在 CIF 時，比 Wu[8] 也有顯著的改善，且

省去的時間幾乎都維持在一定水準，不會因為視訊序列(video sequence)的性質或解析度大小而有很大的不同。

五、結論與未來研究方向

在 H.264 的編碼中，因為畫面內模式出現機率不高，目前大多數研究都著重在跳過畫面內模式。當場景轉換時，兩種模式立場會相反，我們提出的偵測演算法，利用將畫面 downsample 與計算亮度直方圖的差在判斷畫面轉換時達到不錯的效果。當畫面轉換伴隨著轉場特效時，我們方法雖然也可偵測出場景轉換，但此時兩種模式所佔比例仍可能相當接近，這種情況會犧牲掉不少位元。未來的研究中，為解決此種情況，期望利用多組門檻值偵測此情況，並將畫面間到畫面內的編碼流程反轉，以增加編碼效能。

而本論文所提出快速模式決定演算法則透過設定門檻值進行省略掉某些模式計算的策略，並加上動態調整門檻值的方法，可以減少模式決定所花費的時間，提升編碼的速度平均達 35%，而且整個演算法的流程並不複雜，方法簡單又能達到不錯的效果。

由於我們的演算法是修改 H.264 官方參考軟體 JM14.2，因此對於底層移動估測(motion estimation)的計算，是呼叫原本參考軟體的模組，因此我們的方法在這個實驗的實作上，移動估測這方面並沒有省去時間，如果底層移動向量的計算模組可以再改寫成配合我們的演算法，那真正的計算時間可以再省去更多，之後的研究可以朝這方面再繼續改進。

未來的研究方向也希望可以朝著把我們的演算法及 H.264 參考軟體改寫到嵌入式的平台上，因為 JM 這個參考軟體是為了研究實驗用，其速度太慢，如果直接在嵌入式平台上執行是不可行的，要移植必須做非常大幅度的修改，期許未來我們能夠完成在嵌入式平台上可以達成即時編/

解碼的 codec。

參考文獻

- [1] I. Choi, J. Lee, B. Jeon, "Fast coding mode selection with rate-distortion optimization for MPEG-4 part-10 AVC/H.264," IEEE Trans. on Circuits and Systems for Video Technology, vol. 16, no. 12, pp. 1557-1561, Dec. 2006.
- [2] T. J. Kim, J. E. Hong, J. W. Suh, "A Fast Intra Mode Skip Decision Algorithm Based on Adaptive Motion Vector Map," IEEE Trans on Consumer Electronics, vol. 55, no.1, pp 179-184, Feb, 2009.
- [3] P. J. Lee, Y. J. Shin, "Fast Inter-Frame Coding with Intra Skip Strategy in H.264 Video Coding," IEEE Trans on Consumer Electronics, vol. 55, no.1, pp 158-164, Feb, 2009.
- [4] Z. Liu, L. Shen, Z. Zhang, "An Efficient Intermode Decision Algorithm Based on Motion Homogeneity for H.264/AVC," IEEE Trans. on Circuits and Systems for Video Technology, vol. 19, no. 2, pp.128-132, Feb. 2009.
- [5] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, "Fast Mode Decision Algorithm for Intra prediction in H.264/AVC, Video Coding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 15, no. 7, pp. 813-822, July 2005.
- [6] I. E. G. Richardson, "H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia", John Wiley & Sons, Ltd., 2003
- [7] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Trans. on Circuits and Systems for Video Technology, vol.13, no.7, pp.560-576, JULY 2003.
- [8] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C.C. Ko, "Fast inter-mode decision in H.264/AVC video coding," IEEE Trans. Circuit Syst. Video Technol., vol. 15, no. 6, pp. 953 - 958,

Jun.2005.

- [9] P. Yin, H. Y. C. Tourapis, A. M. Tourapis, J. Boyce, "Fast Mode Decision and Motion Estimation for JVT/H.264," in Proc. Int. Conf. Image Process., 2003, pp. 853 - 856.
- [10] A. C. W. Yu, G. R. Martin, H. Park, "Fast Inter-Mode Selection in the H.264/AVC Standard Using a Hierarchical Decision Process" IEEE Trans. on Circuits and Systems for Video Technology, vol. 18, no. 2, pp. 186-195, Feb. 2009.