

螢幕內容編碼之快速可調式色彩轉換

Fast Adaptive Color Transform for Screen Content Coding

涂日昇 張耀仁 林敬傑 林俊隆
Jih-Sheng Tu, Yao-Jen Chang, Ching-Chieh Lin, Chun-Lung Lin

中文摘要

螢幕內容編碼技術(SCC)是一個基於高效能視訊編碼技術(HEVC)的擴展標準。這個編碼標準可以針對一些螢幕分享應用，諸如雲端遊戲、螢幕畫面分享、遠端桌面操作及無線顯示內容等，提供更有效率的編碼壓縮方法。可調式色彩轉換(Adaptive Color Transform; ACT)是螢幕內容編碼技術其中一項編碼工具，被用於冗餘值(residue)的色彩空間轉換，它能夠有效地降低色彩空間組成之間的相關性，並因此而提升編碼效果。然而，要評估目前這個編碼單元(coding unit)是否該啟用ACT需要額外的碼率失真函數(rate-distortion optimization)計算，因而花費大量的編碼時間。我們將在本文裡介紹一個更有效率的ACT編碼流程，它藉由判斷目前這個編碼單元的尺寸大小，來決定是不是要提前終止轉換單元(transform unit)的分割流程。根據實驗模擬結果顯示，我們提出的方法能夠在幾乎不犧牲編碼效益的情況下，最多縮短了8%的編碼時間。

Abstract

Screen Content Coding(SCC) is an extension of High Efficiency Video Coding(HEVC). It targets to further improve the coding efficiency for the applications, such as cloud gaming, screen sharing, remote computer desktop control and wireless display of content. Adaptive color transform (ACT) is one of the new coding tools in SCC extension. ACT is a color space transform applied to residual data of a coding unit. It could effectively de-correlate color components of pixel samples and accordingly improve the resulting coding performance. However, when ACT is enabled, the encoder currently needs to perform additional rate-distortion checks which are extremely time-consuming. In this article, we propose an optimal ACT coding flow for SCC. The proposed methods can reduce the encoding complexity by early terminating the recursive transform unit(TU) splitting. The early termination is activated while the current CU size is greater than a pre-defined threshold. The experimental results show that the proposed methods can reach up to 8% encoding time saving with negligible coding gain loss.

關鍵詞(Key Words)

高效能視訊編碼標準 (High Efficiency Video Coding ; H.265 / HEVC)

螢幕內容編碼擴展 (Screen Content Coding extension ; SCC extension)

可調式色彩轉換 (Adaptive Color Transform ; ACT)

1 · 前言

為了因應高畫質(Full High Definition; Full HD)影音視訊應用日漸蓬勃的發展，國際視訊編碼標準組織「動態影像專家群組(Moving Picture Experts Group; MPEG)」[1]及「國際電信聯盟電信標準化部門(International Telecommunication Union Telecommunication Standardization Sector; ITU-T)」[2]，合作推出了一個共同標準—高效能視訊編碼標準(High Efficiency Video Coding; H.265 / HEVC)[3][4]，並且於2013年發布了這個標準的第一個版本(HEVC version 1)。H.265 / HEVC不僅能滿足Full HD的應用需求，而且適用範圍更擴展到超高畫質(Ultra High Definition; UHD)這一類4k、8K的超高解析度視訊規格，憑藉著H.265 / HEVC對視訊影片的高壓縮率，相較於以往，使用者更能透過網路於各式的終端設備觀看高品質的影片，甚至也能夠在視訊會議及螢幕共享的過程中顯示高品質的影像。

螢幕內容係指由圖案、文字或動畫所成像(render)的視訊，而不是由攝影機所擷取的視訊場景。MPEG和ITU-T為了專注於強化螢幕內容共享這一類服務的應用情境，特別以H.265 / HEVC為基礎開發了一個擴展標準，名為螢幕內容編碼技術擴展(Screen Content Coding extension; SCC extension)。SCC針對螢幕內容視訊的編碼壓縮，可以取得比H.265 / HEVC更為突出的編碼效能增進，平均而言有高達百分之五十的效能提升。SCC的應用實例則可以包含螢幕鏡射(screen mirroring)、雲端遊戲(cloud gaming)、無線顯示內容(wireless display of content)、遠端電腦存取時的顯示(displays generated during remote computer desktop access)及螢幕共享(screen sharing)—例如視訊會議之即時螢幕共享。

在本篇論文裡，我們將在第二章對SCC的技術做概略性的簡單介紹，而在這些技術裡面，本篇論文選定了其中一項名為可調式色彩轉技術(Adaptive Color Transform; ACT)做更詳細的說明，並且於第三章提出我們針對ACT配合轉換單元(Transform Unit; TU)切割流程所發明的編碼優化流程，這個方法可以有效地降低

ACT作用在YUV色彩空間時所花費的冗餘時間，而根據第四章我們提供本方法實作後的實驗數據，證實本方法可以在近乎不犧牲畫質的情況下，節省5%~8%的編碼複雜度。

2 · 螢幕內容編碼技術之簡介

我們將在本章節簡介目前SCC視訊標準最重要的三項編碼技術，它們分別是畫面內區塊複製技術(Intra-Block Copy; IBC)、調色盤編碼技術(Palette-Based Coding; PBC)以及ACT等三大技術，如果想更深入地瞭解螢幕內容編碼技術，建議請參閱標準文件[5]。

2.1 畫面內區塊複製技術

有鑒於現今視訊串流的壓縮結果，I-畫框(I-frame)的資料量相較於P畫框(P-frame)而言會高出不少，因此當我們在傳輸一些高位元率(bit-rate)的視訊串流時，有可能會因為I-frame傳輸花了比較多的時間而造成延遲；對此，螢幕內容編碼技術針對I-frame的壓縮率新增了改善的方法，命名為畫面內區塊複製模式(Intra-Block Copy mode; IBC mode)或是當前畫面參照(Current Picture Referencing; CPR)；過去傳統的畫面間預測(inter prediction)技術是從已經編碼完的frame去尋找預測區塊(predictor)，但是IBC的做法則是讓編碼區塊在尋找預測區塊的時候，是在自己的畫面內做搜尋，兩個技術的差異如圖1所示：

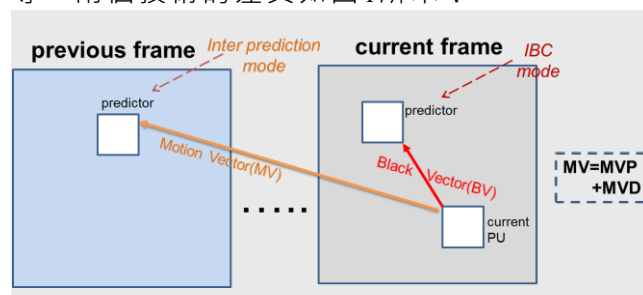


圖 1 畫面內預測和畫面間預測的差異比較

2.2 調色盤編碼技術

PBC的基本概念是找出目前要編碼的單位區塊(Coding Unit; CU)裡，最主要的幾個顏色。根據這些找出來的主要顏色(major color)，對該顏色出現的次數，進行排序建立出主要顏色

表。接下來，編碼器將會針對這些主要顏色進行編號，出現次數越多次的顏色，就會給與比較小的索引值，並且用比較少的 bit 數來表示，反之亦然。

把CU裡面每個像素經過主要顏色表轉換後，我們可以得到一個由像素值轉換而成的索引圖(index map)，這個轉換流程的概念請參考圖2。另外，為了增加壓縮這個索引圖的效率，PBC不是直接編碼壓縮這張索引圖，而是使用了可變長度編碼的技巧，去減少紀錄一些重複的片段，因此可以大幅度地增加了PBC的編解碼效能，我們簡介這些方法如下：

- **索引模式(index mode)**：index mode會編一個索引值以及一個長度值，用來表示該索引值會連續重複出現該長度的次數。
- **向上複製模式(copy above mode)**：copy above mode會需要編碼一個長度值，用來表示在這個長度內的索引值都和它們上一列那個像素位置的索引值相同。
- **跳脫模式(escape mode)**：escape mode算index mode的一種變型，我們在建立主要顏色表的時候，會把一些無法被歸類的像素值通通當成跳脫值(escape)，當解碼器發現解碼出來的索引值等於跳脫值的時候，就知道現在是escape mode，所以就會知道後面要解碼的不是一個長度值，而是這個像素的值。

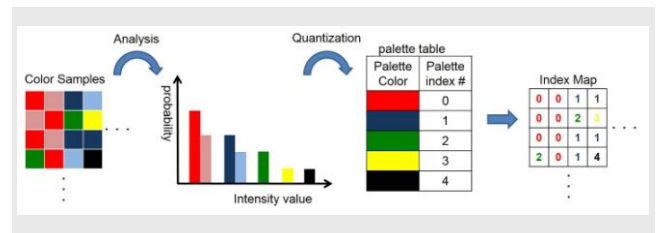


圖 2 調色盤編碼技術之色彩索引轉換示意圖

2.3 可調式色彩轉換技術

SCC允許編碼器對經過編碼預測後的冗餘值(residue)，在進行轉換(transform)及量化(quantization)之前先作一次色彩空間轉換，並且在CU的檔頭(header)標記這個CU是否要作該色彩空間轉換，這樣的流程就是ACT的主要精神；在目前的SCC標準裡ACT使用的色彩空間轉換公式如式(2.3-1)所示：

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} / 4 \quad (2.3-1)$$

透過式(2.3-1)的色彩空間轉換，有時候可以增進transform和quantization的效果進而提高編碼壓縮率；然而我們從圖3 ACT的編碼流程圖可以看得出來，每次評估是否啟用ACT的時候需要額外的碼率失真函數(Rate Distortion Optimization；RDO)，RDO被用來評估碼率失

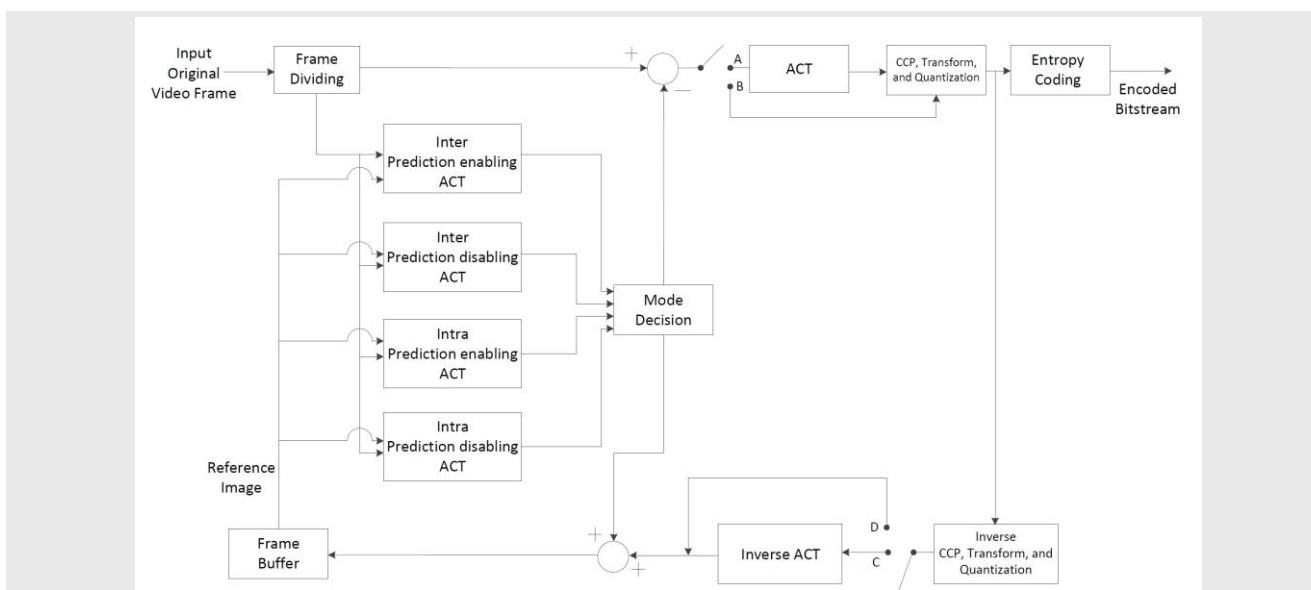


圖 3 ACT編碼流程圖

真(Rate Distortion ; RD)的成本。但是這些評估過程可能會增加編碼複雜度以及編碼時間。再者，當經過色彩空間轉換後，如果不能有效地提升transform和quantization的結果，那麼此時ACT可能就不是必要的。在這種情況下，由於執行ACT的成本高於編碼的效益，如果能適當地在某些條件下跳過ACT部分的編碼流程，可在不損及編碼效能的情形下加快編碼速率。

3 · ACT編碼流程優化演算法

3.1 演算法概念發想

根據前文對ACT的介紹，我們可以得知不管是畫面內預測(intra prediction)或是畫面間預測(inter prediction)在決定目前這個CU要不要啟用ACT的時候，都無可避免的會使用到RDO來計算編碼成本，然而我們發現在[6]所提供的實驗數據裡，對於YUV格式的影片來說，ACT無論開啟與否，它所影響的編碼壓縮率其實相當有限，不過對RGB格式的影片來說結果就不是這樣了，我們從這點判斷，如果影片格式屬於高相關性的色彩空間組成(例如RGB格式)，ACT比較能發揮功用，反之，如果影片格式屬於低相關性的色彩空間組成(例如YUV格式)，那麼ACT就顯得不是很必要了。

除此之外，當目前CU(current CU)是一個大尺寸CU的時候，這通常意味著這個CU所包含的區域並沒有太過複雜的圖形紋理，所以這時候有沒有開啟ACT大概也不會有太大的區別，我們也可以推測在這樣的條件下，對這個CU進行TU切割(TU splitting)所能帶來的效益恐怕也會很小，這樣一來等於無形中浪費了編碼運算資源，拖慢SCC的編碼速度。

依上述的觀察，我們可以歸納出下列幾個可能可以有效降低ACT編碼複雜度的手段：

- (1) 如果當前要壓縮的視訊影片，它的色彩空間組成是屬於低相關性的話，略過ACT編碼流程可能是一種可以接受的選擇。
- (2) 在進行ACT編碼的流程裡面，適度的跳過一些大尺寸的CU不讓它們運算TU分割，藉此節省後續計算ACT、跨色

彩組成預測(Cross Component Prediction ; CCP)、transform、quantization、熵編碼(entropy coding)所花費的運算時間(見圖3)。

3.2 所提的ACT編碼流程優化演算法

基於上一小節的推演，我們在這一小節將詳述我們發明的ACT編碼流程優化演算法，而這個演算法只應用於畫面內預測。

我們總共設計了兩個方法並分述如下：

(1) 方法一：

首先，假設現在編碼流程進行到計算ACT開啟時的RD成本這個步驟，那麼我們的方法會先判斷目前CU的色彩空間組成，是屬於高相關性或是低相關性(如何決定相關性高低將在後文介紹)，如果是屬於高相關性的話，則按照原始編碼程序繼續進行後續編碼流程，如果不是的話，我們將確認目前CU的尺寸大小是否小於一個給定的臨界值 T_1 (例如64x64)，若是，則按照原始編碼程序繼續進行後續編碼流程，若否，則提前結束編碼程序，不去計算目前CU開啟ACT所得到的RD成本。

(2) 方法二：

我們在ACT開啟的時候，當目前的編碼程序正要進行TU尺寸的RDO比較評估之前，多加入了一個判斷程序，該判斷程序做的事是確認目前的CU尺寸是不是小於等於一個給定的臨界值 T_2 ，如果小於等於的話，則按照原始編碼程序繼續進行後續編碼流程，如果大於的話，則提前終止計算這個TU尺寸的RD成本。

圖4是方法一加方法二的流程圖，我們說明如下，CU一開始先進入成分相關性分析(component correlation analysis)程序，這個程序會去計算CU內之各個畫素之色彩成分的相關性得到一個量化值，如果該值低於一個預先決定好的值，則判定這個CU它的色彩空間組成屬於低相關性，因此，流程進入方法一的判斷步驟，並決定是否提前終止編碼流程；然而，若是目前CU的色彩空間組成被判定是屬於

高相關性，則往下繼續進行編碼流程。

程序概略模式決定 (rough mode decision) 採用簡單、低複雜度編碼成本的方法從畫面內預測模式 (Intra Prediction Modes ; IPM) 裡，選出數個低編碼成本高編碼品質的IPM，例如使用絕對轉換誤差和 (Sum of Absolute Transform Distortion ; SATD) 來決定出各個IPM的低複雜度編碼成本。而在程序碼率失真函數模式決定 (RDO based mode decision) 裡面，再針對這些挑選出來的IPM所產生的residue進行ACT、CCP、transform、quantization、entropy coding等計算得到精細的RD成本，只不過依照目前CU的尺寸大小不同，這樣的RD成本計算會因為有多種TU分割的可能而需要執行好幾次，比方說一個尺寸大小是32x32的預測單元 (Prediction unit ; PU) 它可能可以有32x32、16x16、8x8、4x4等好幾種TU分割的結果，程序TU尺寸決定 (TU size decision) 就是在求出能獲得最佳RD成本的TU尺寸。

由於TU尺寸的最佳化會大幅的消耗編碼運算，即增加編碼複雜度，因此，我們在進行程序TU尺寸決定之前先以一個判斷式來決定目前CU要不要進行TU分割，如果CU尺寸大於一個給定的臨界值 T_2 ，就跳過程序TU尺寸決定，藉此提高編碼速率；再者，CU尺寸大於臨界值 T_2 ，其實代表CU的內容可能不會太過複雜，舉例來說，像是非物件邊界、非動態物件的區域或是平滑的圖像等等。而決定跳過程序TU尺寸決定的時候，這時候的TU尺寸將會設定為目前CU尺寸和TU允許的最大尺寸，這兩者中比較小的那一個。

最後的色度模式決定 (chroma mode decision) 是依據已決定的預測模式來使色度預測 (chroma prediction) 產生色度PU (chroma PU) 及對應的色度TU (chroma TU)，例如當色度取樣格式為4:2:0，且亮度TU的尺寸為32x32，則此時的色度TU的尺寸將為16x16。

完成上述流程我們將得到目前CU在開啟ACT的情況下RD成本，這個RD成本將與其它編碼模式所計算出來的RD成本競爭，挑選出最好的編碼模式。

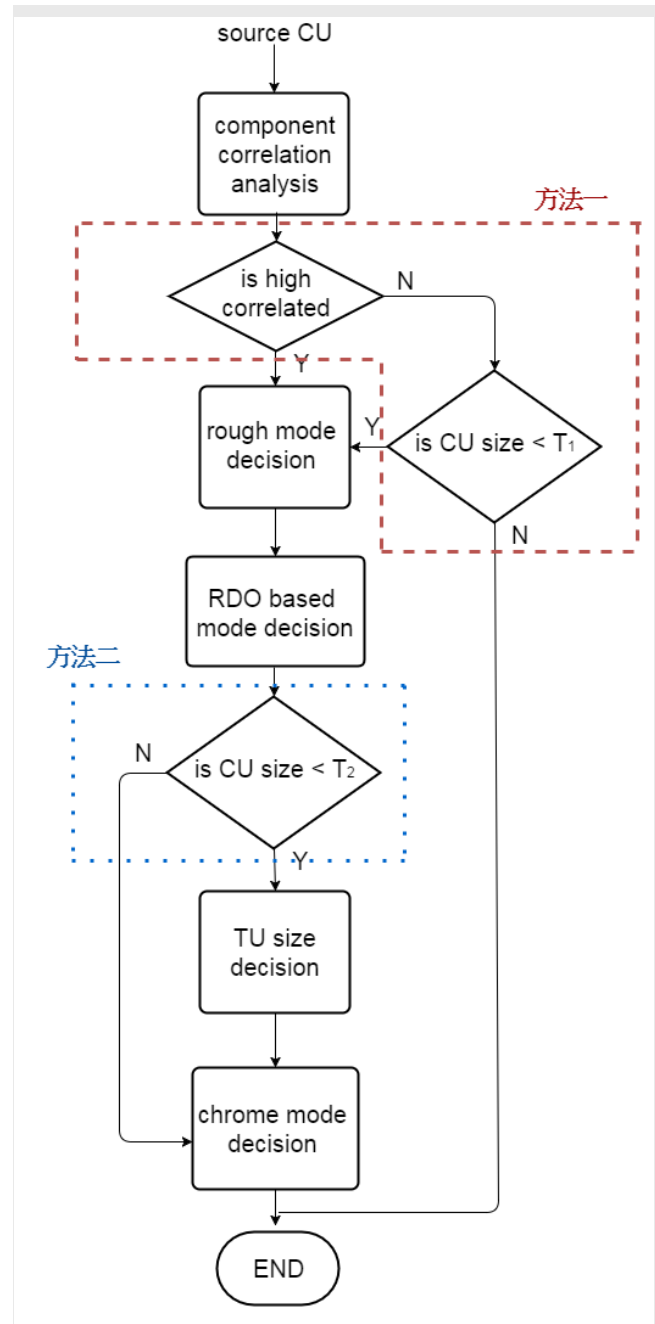


圖 4 所提的ACT編碼流程優化演算法 - 方法一加方法二流程圖

4. 實驗結果

我們實作本論文的演算法在SCC視訊標準所提供的參考軟體螢幕內容編碼測試模範第四

版(Screen Content Coding Test Model 4.0 ; SCM-4.0) · 而實驗模擬手段也依照標準組織提供的通常測試條件(Common Test Conditions ; CTC)[7] · 進行必要的測試 · 我們雖然有完成全部的測試的條件 · 但礙於本文的篇幅 · 我們僅於下文中列出測試條件為基於色彩取樣格式為4 : 4 : 4和4 : 2 : 0 · 進行有損(lossy)編碼且全畫面IBC(full-frame IBC)開啟的情況下 · 執行三種編碼策略 · 分別是全畫面內(All Intra ; AI) · 隨機存取(Random Access ; RA)和低延遲雙向預測(Low-Delay B ; LD-B)的實驗數據 ; 而受測影片也來自於標準組織 · 總共有YUV和RGB兩種色彩空間格式(色彩空間格式RGB沒有色彩取樣格式為4 : 2 : 0的影片) · 這些影片包含單純的螢幕內容視訊 · 或是螢幕內容混合一般攝影機拍攝的視訊影片等等 · 關於SCM和受測影片的更詳細描述請參閱CTC[7] ·

本論文提供方法二及方法一加方法二的實驗測試數據 · 參數的設定為 T_1 固定為尺寸大小64x64 · 另外測試 T_2 等於尺寸大小32x32以及尺寸大小16x16這兩種設定 · 由於我們發明的演算法在RA和LD-B兩個測試條件下 · 對編碼時間和編碼品質不會有顯著的變化 · 因此我們只在表1至表6列出方法二以及方法一加方法二在AI測試條件下 · 改變 T_2 的值所影響的BD-rate與編碼時間 ·

而關於我們對程序成分相關性分析(component correlation analysis)的實作方面 · 在本次的實驗模擬是直接把YUV色彩空間格式的影片當作低相關性 · 然後把RGB色彩空間格式的影片當作高相關性來處理 ·

在表1至表6裡 · 最左邊的欄位提供受測影片的內容分類 · 色彩空間格式以及畫面解析度等資訊 · 右方的欄位則顯示每個色彩空間組成在該影片分類下的BD-rate變化 · 其中如果該值為負則表示編碼效益(coding gain)獲得提升 · 反之如果該值為正則代表coding gain下降 ; 表中的Enc Time[%]和Dec Time[%]則表示我們的演算法相較於SCM-4.0在編碼時間和解碼時間各有多少百分比的變化 · 亦即與原始SCM-4.0相比 · 我們的方法減少或增加了多少編解碼時間 ·

表 1 方法二於 T_2 設定為尺寸大小32x32 · 色彩空間格式為4 : 4 : 4的時候 · AI的實驗結果

	All Intra		
	GY	B/U	RV
RGB, text & graphics with motion, 1080p & 720p	-0.1%	0.0%	-0.1%
RGB, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%
RGB, Animation, 720p	0.0%	0.0%	0.0%
RGB, camera captured, 1080p	0.0%	0.0%	0.0%
YUV, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.1%
YUV, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%
YUV, Animation, 720p	0.0%	0.0%	0.0%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%
Enc Time[%]	96%		
Dec Time[%]	100%		

表 2 方法二於 T_2 設定為尺寸大小16x16 · 色彩空間格式為4 : 4 : 4的時候 · AI的實驗結果

	All Intra		
	GY	B/U	RV
RGB, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.0%
RGB, mixed content, 1440p & 1080p	0.1%	0.1%	0.1%
RGB, Animation, 720p	0.1%	0.1%	0.1%
RGB, camera captured, 1080p	0.1%	0.0%	0.1%
YUV, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.0%
YUV, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%
YUV, Animation, 720p	0.0%	0.2%	0.1%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%
Enc Time[%]	93%		
Dec Time[%]	100%		

表 3 方法一加方法二於 T_2 設定為尺寸大小32x32 · 色彩空間格式為4 : 4 : 4的時候 · AI的實驗結果

	All Intra		
	GY	B/U	RV
RGB, text & graphics with motion, 1080p & 720p	-0.1%	0.0%	-0.1%
RGB, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%
RGB, Animation, 720p	0.0%	0.0%	0.0%
RGB, camera captured, 1080p	0.0%	0.0%	0.0%
YUV, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.1%
YUV, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%
YUV, Animation, 720p	0.0%	0.1%	0.1%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%
Enc Time[%]	95%		
Dec Time[%]	100%		

表 4 方法一加方法二於 T_2 設定為尺寸大小16x16 · 色彩空間格式為4 : 4 : 4的時候 · AI的實驗結果

	All Intra		
	G/Y	B/U	R/V
RGB, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.0%
RGB, mixed content, 1440p & 1080p	0.1%	0.1%	0.1%
RGB, Animation, 720p	0.1%	0.1%	0.1%
RGB, camera captured, 1080p	0.1%	0.0%	0.1%
YUV, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.0%
YUV, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%
YUV, Animation, 720p	0.0%	0.2%	0.2%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%
Enc Time[%]	92%		
Dec Time[%]	100%		

表 5 方法一加方法二於 T_2 設定為尺寸大小32x32 · 色彩空間格式為4 : 2 : 0的時候 · AI的實驗結果

	All Intra		
	G/Y	B/U	R/V
YUV, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.0%
YUV, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%
YUV, Animation, 720p & 768p	0.0%	0.0%	0.0%
Enc Time[%]	100%		
Dec Time[%]	100%		

表 6 方法一加方法二於 T_2 設定為尺寸大小16x16 · 色彩空間格式為4 : 2 : 0的時候 · AI的實驗結果

	All Intra		
	G/Y	B/U	R/V
YUV, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.0%
YUV, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%
YUV, Animation, 720p & 768p	0.0%	0.0%	0.0%
Enc Time[%]	100%		
Dec Time[%]	100%		

從實驗結果可以發現 · 本論文所提出的編碼加速方法並不會造成coding gain大幅衰退 · 幾乎是可以忽略的程度 ; 在編碼時間方面 · 方法二大致可以節省4%到7%的編碼時間 · 方法一加方法二的話則可以節省5%到8%的編碼時間 · 又因為色彩取樣格式為4 : 2 : 0的影片只有

YUV色彩空間格式 · 所以本演算法沒能對它們產生作用 ·

5 · 結論

本論文探討在SCC標準裡面 · ACT這項重要的編碼工具 · 對不同的色彩空間而言有不同的適用性 · 我們藉由這樣的特性 · 發明一套ACT編碼流程優化演算法 · 藉由一些條件判定 · 減少RDO確認的計算量 · 並保留需要執行ACT的地方 · 以避免coding gain損失過大 ·

根據實驗結果所示 · 我們所提出的方法確實有效地在加速整體編碼速度之餘 · 依然保留了幾乎是全部的coding gain ; 雖然隨著採用的方法而略有差異 · 本論文所提出的演算法約略能帶來5%到8%的編碼加速 ·

同時 · 本論文提出的方法也保留了設計上的彈性 · 像是一些門檻值(T_1 · T_2)的設定 · 實務上也可以依據目前編碼的狀況去進行分析 · 動態地調整這些門檻值 · 另外 · 在演算法流程裡提到的成分相關性分析 (component correlation analysis) 程序 · 目前是把輸入的視訊影片色彩空間格式當成已知資訊 · 未來 · 如何在色彩空間格式未知的情形下 · 簡潔的計算出所謂的色彩空間組成相關性 · 也將是日後改良本演算法的方向 ·

參考文獻

- [1] ISO/IEC JTC1/SC29 WG11, http://www.iso.org/iso/iso_technical_committee?commid=45316
- [2] ITU-T SG16, <http://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/default.aspx>
- [3] Mathias, “High Efficiency Video Coding,” Springer-Verlag Berlin Heidelberg, 2015.
- [4] Vivienne Sze, Madhukar Budagavi, and Gary J. Sullivan, “High Efficiency Video Coding (HEVC),” Springer International Publishing, 2014.
- [5] R. Joshi, S. Liu, G. Sullivan, G. Tech, Y. K. Wang, J. Xu, and Y. Ye, “High Efficiency Video Coding (HEVC) Screen Content

Coding: Draft 6,” JCTVC-W1005, Feb. 2016.

[6] P. Lai, S. Liu, and S. Lei, “AHG6: Coding performance of SCC tools and suggested common test conditions,” 20th JCT-VC meeting, JCTVC-T0080, Geneva, Switzerland, Feb. 2015.

[7] H. Yu, R. Cohen, K. Rapaka, and J. Xu , “Common Test Conditions for Screen Content Coding,” 20th JCT-VC meeting, JCTVC-T1015, Geneva, Switzerland, Feb. 2015.

作者簡介

涂日昇



現任職於工研院資通所視訊編碼核心技術部工程師。專長為視訊編碼。

[E-mail:sunrisejstu@itri.org.tw](mailto:sunrisejstu@itri.org.tw)

張耀仁



國立中央大學通訊工程系博士，現任工研院資通所視訊多媒體通訊技術組視訊編碼核心技術部資深工程師。專長為視訊壓縮訊號處理、通訊訊號處理。目前從事視訊標準制定。

[E-mail : britpablo@itri.org.tw](mailto:britpablo@itri.org.tw)

林敬傑



現任職於工研院資通所視訊編碼核心技術部資深工程師。專長為視訊編碼。

[E-mail:JackLin@itri.org.tw](mailto:JackLin@itri.org.tw)

林俊隆



國立清華大學資訊工程系博士，現任工研院資通所視訊多媒體通訊技術組視訊編碼核心技術部副經理。專長為多媒體視訊傳輸與編碼、影像處理、圖形辨識、無線感知器網路等。

[E-mail:Chunlung@itri.org.tw](mailto:Chunlung@itri.org.tw)