

國立交通大學

資訊科學與工程研究所

碩 士 論 文



研 究 生：吳牧軒

指 導 教 授：彭文孝 教授

中 華 民 國 一 百 零 二 年 二 月

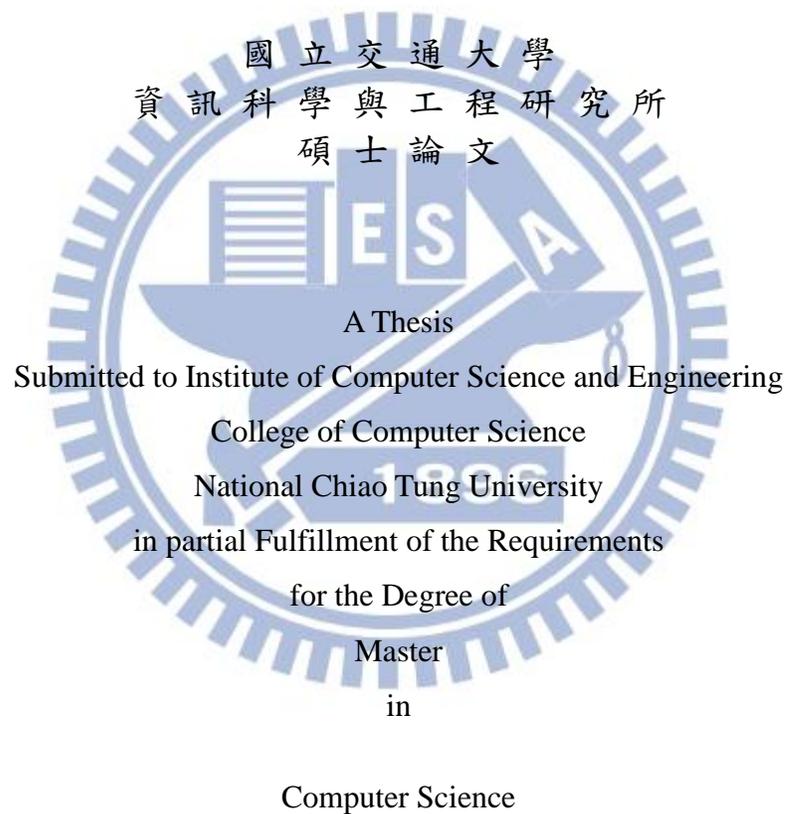
視窗影像編碼研究
Study on Screen Content Coding

研究生：吳牧軒

Student：Mu-Hsuan Wu

指導教授：彭文孝

Advisor：Wen-Hsiao Peng



February 2013

Hsinchu, Taiwan, Republic of China

中華民國一百零二年二月

視窗影像編碼研究

研究生：吳牧軒

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

視窗影像是泛指電腦所生成圖像，其中包含文字、電腦繪圖等，且伴隨自然影像。傳統混合式編碼為主的壓縮技術並無法有效壓縮此類影像，因此本文將視窗影像分成文字像素及圖像像素，並針對圖像像素做不同演算法比較。圖像像素可分為電腦繪圖，例如漸層繪圖、紋理樣式等，以及實際拍攝下的自然影像。所使用編碼演算法中，有針對漸層的二元一次方程式，或是自然影像的傳統離散餘弦轉換。但圖像像素並非完整區塊，因此離散餘弦轉換時，必須將空缺填補或使用針對非完整區塊設計的離散餘弦轉換，其為匹配演算法(Matching Pursuit)及外型可調式離散餘弦轉換(Shape Adaptive Discrete Cosine Transform, SADCT)。客觀數據下使用匹配演算法，不管圖像像素為漸層或是自然影像下有不錯壓縮效果，且可視為編碼端問題，不需更改解碼端設計。但圖像像素為紋理樣式時無好處，其中可能為使用的基底對於紋理樣式無法表示。另外採用外型可調式離散餘弦轉換演算法，由於會根據像素個數，採用不同長度的離散餘弦轉換，因此造成主觀上不連續之問題，且大區塊下會有更明顯的失真現象。針對漸層的圖像像素，由於二元一次方程式可產生相似的漸層背景，因此主觀上能得到好處。至於圖像像素為紋理樣式時，採用調色盤(Palette based)這類傳送基底顏色，並編碼各像素所對應的基底顏色，高位元率時能有較佳表現。

Study on Screen Content Coding

Student : Mu-Hsuan Wu

Advisor : Wen-Hsiao Peng

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT

Screen contents are synthesized images generated by computer, which are generally composed by computer generated text, graphics, and natural images. Traditional hybrid video coding is inefficient to deal with such contents. To achieve a higher coding efficiency, this thesis classifies screen contents into text and pictorial pixels, and aims at comparing the performances of different methods for coding pictorial pixels. There are two kinds of pictorial pixels: one of which is computer graphics such as color gradient, graphic texture, and etc., and the other is natural images. The polynomial function is proposed mainly for dealing with color gradient while traditional discrete cosine transform is suitable for natural image. However, pictorial pixels cannot compose an entire image, and those non-pictorial pixels, which are viewed as holes, need to be filled before such incomplete image can be processed by discrete cosine transform. Other than that, this thesis also investigates other methods, such as matching pursuit and shape adaptive discrete cosine transform (SADCT), for incomplete images. The matching pursuit performs adequately for both color gradient and natural image, and is friendly for hardware design of the core transformation at decoder. However, the bases of matching pursuit may sometimes not suitable for characterizing the signal characteristics of graphic texture. A 2-D separable SADCT transform incorporates with variable-length 1-D DCT bases according to the horizontal and vertical lengths of input pixels. It will cause subjective discontinuity and will have distinct distortion on big fragment. The polynomial function works well on color gradient due to the capability of the polynomial function for generating similar gradient background. On the other hand, graphic texture works well at high rate by using palette-based scheme.

誌謝

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、如何思考問題以及解決問題的方法，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，讓我能順利的完成碩士論文及未來的生命中也受益無窮。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，我也要感謝吳崇豪學長和陳俊吉學長在我的研究上擔任了啟蒙的指導，不僅在HEVC和SCC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、與詹家欣 博士、李宗霖、黃嘉彥、曾于真、與陳孟傑，一步步帶領我進入這個專業的領域，給予適時的意見與建議；感謝我的同學陳彥宇，不管在課業或是研究上，都可以給予幫助；感謝我的學弟朱弘正、吳昱興、王信硯，提供我生活及研究上所有的幫助。

最後，我要感謝我的父母—吳仁宗 先生、陳月慧 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。感謝我的弟弟—吳牧宸，不斷的鼓勵我繼續向前。感謝我的女友—朱文歆，適時的提供我意見，以及這幾年來辛苦的陪伴、體貼和關心。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

中文摘要	i
ABSTRACT	ii
誌謝	iii
目錄	iv
圖目錄	vi
表目錄	viii
第一章、緒論	1
1.1 背景介紹	1
1.2 研究貢獻	2
1.3 論文編排	2
第二章、相關研究	3
2.1 基於形狀之編碼方式 (Shape based)	3
2.2 基於調色盤之編碼方式 (Palette based)	4
2.3 分層架構之編碼方式 (Layer based)	6
2.3.1 填補空缺之演算法 (Hole filling)	6
2.3.2 Matching pursuit 直接計算 DCT 係數	7
2.4 基於字典架構之編碼方式 (Dictionary based)	9
2.5 摘要	10
第三章、研究方法	11
3.1 視窗影像特性及 BCIM 之分析	11
3.1.1 Transform 對視窗影像影響之分析	11
3.1.2 使用 K-means & BCIM 取代量化及嫡編碼設計	14
3.1.3 分析 Intra prediction 對 BCIM 之影響	14
3.2 演算法概念	20
3.3 確定性關係表示特殊平面	21
3.3.1 針對線性平面的分層切割演算法	22

3.3.2	Entropy Coding	23
3.4	統計上關係表示特殊平面	24
3.4.1	Shape-adaptive DCT (SADCT)	24
3.4.2	Entropy Coding	28
第四章、實驗結果與討論	29
4.1	確定性關係(漸層平面)之實驗結果	29
4.1.1	切割演算法中不同閾值之選擇	29
4.1.2	BCIM 和漸層演算法之比較	29
4.2	統計上關係(SADCT)之實驗結果	36
4.2.1	SADCT 表示前景文字像素	36
4.2.2	SADCT 表示背景圖像像素	37
4.3	SADCT 的問題討論	47
第五章、結論	50
參考文獻		



圖目錄

圖 1：Class F 視窗測試影像(a) <i>BasketballDrillText</i> ;(b) <i>ChinaSpeed</i> ;(c) <i>SlideEditing</i> ;(d) <i>SlideShow</i> ;	1
圖 2：視窗影像(a) 漸層背景;(b) 自然影像背景	2
圖 3：TS 流程圖	3
圖 4：根據形狀編碼之架構	3
圖 5：切割方塊流程圖	4
圖 6：BCIM 之架構	4
圖 7：Context model 所參考的像素位置	5
圖 8：Context model 的選擇方法	5
圖 9：調整索引值的方法	5
圖 10：MRC 架構圖	6
圖 11：參考鄰近資料的填補方法	7
圖 12：POCS 概念圖	7
圖 13：Matching pursuit 演算法	9
圖 14：將二維視窗影像區塊掃描成一維陣列	9
圖 15：字典編碼概念	9
圖 16：視窗影像區塊(a) 作業視窗的圖標;(b) 較複雜的文字;(c) 圖像複合簡單文字;(d) 電子線路圖;	10
圖 17：綠色標示為 Screen content 特性的區域(Class F – China Speed)	12
圖 18：綠色標示為 Screen content 特性的區域(Class F – Slide Editing)	12
圖 19：自然影像特性的區域	12
圖 20：Screen content 特性的區域	13
圖 21：原始像素值(Original)間的 Correlation coefficient matrix(a) Transform block;(b) Screen content block;	13
圖 22：餘值(Residual)間的 Correlation coefficient matrix(a) Transform block;(b) Screen content block;	13
圖 23：有無預測(Prediction)時 BCIM mode 之主觀差異	15
圖 24：區塊內像素(a)原始像素值(Original);(b)預測值(DC prediction);(c)重建值	15
圖 25：有預測及無預測其區塊內像素值的分佈圖	17
圖 26：預測和無預測在不同重建值個數下其誤差量變化	19
圖 27：預測和無預測在不同重建值個數下其位元花費	19
圖 28：視窗影像分成文字像素及圖像像素(a) 確定性關係表示圖像像素;(b) 統計上關係表示圖像像素;	20
圖 29：演算法流程	21
圖 30：加入特殊平面的編碼架構	21
圖 31：區塊內像素值的直方圖	23
圖 32：Linear Consensus (LC)的切割方式	23
圖 33：SADCT 流程圖	24
圖 34：Normalize SADCT 對 DC 的影響 (座標 X、Y 為像素之位置)	27
圖 35：分離 DC 流程圖(正轉換)	28

圖 36 : 分離 DC 流程圖(逆轉換)	28
圖 37 : 左圖為原本掃描方式、右圖為修改過之掃描方式	28
圖 38 : HEVC 中新討論的測試影像(a) <i>sc_cad_waveform</i> ;(b) <i>sc_cg_twist_tunnel</i> ;(c) <i>sc_map</i> ;(d) <i>sc_pcb_layout</i> ;(e) <i>sc_ppt_doc_xls</i> ;(f) <i>sc_ppt_doc_xls</i> ;(g) <i>sc_programming</i> ;(h) <i>sc_video_conferencing_doc_sharing</i> ;(i) <i>sc_web_browsing</i> ;(j) <i>VenueVu</i> ;	32
圖 39 : SlideEditing 的主觀視覺(a) Original;(b) HM 7.0: 14308.00 kbps;(c) HM+BCIM: 14705.96 kbps;(c) HM+Linear plane: 12297.26 kbps;	33
圖 40: Gradient 在 Qp37 之主觀視覺(a) Original;(b) HM7.0: 0.26 bits/pixel;(c) HM+BCIM: 0.27 bits/pixel;(d) HM+BCIM: 0.21 bits/pixel (e) HM+LinearPlane(LC): 0.24 bits/pixel;(f) HM+LinearPlane(GT): 0.24 bits/pixel;	34
圖 41 : (a) White;(b) Gradient;	34
圖 42 : 漸層平面相關實驗的 Rate distortion curves	35
圖 43 : 漸層平面演算法中不傳送 a、b、c 參數的 Rate distortion curves	35
圖 44 : SlideEditing 文字部分	36
圖 45 : (a) <i>Natural_Ocean</i> ;(b) <i>Natural_cloud</i> ;	38
圖 46 : <i>Natural_Ocean</i> 及 <i>Natural_cloud</i> 之背景使用 HM 壓縮的 Rate distortion curves... ..	39
圖 47 : 自然背景影像實驗的 Rate distortion curves	39
圖 48 : <i>Natural_cloud</i> 在 Qp37 之主觀視覺(a) Original;(b) HM+BCIM 7.0: 0.31 bits/pixel;(c) HM+SADCT: 0.32 bits/pixel;(c) HM+Matching Pursuit: 0.32 bits/pixel;.....	40
圖 49 : (a) <i>Natural_cloud(black)</i> ;(b) <i>Natural_cloud(white)</i> ;	41
圖 50 : 不同文字亮度實驗的 Rate distortion curves	41
圖 51 : Qp37 時 Matching Pursuit 於 <i>Natural_cloud(black)</i> 及 <i>Natural_cloud(white)</i> 所使用的區域(綠色區域)	42
圖 52: Qp37 時 SADCT 於 <i>Natural_cloud(black)</i> 及 <i>Natural_cloud(white)</i> 所使用的區域(綠色區域)	42
圖 53 : Qp37 時 BCIM 於 <i>Natural_cloud(black)</i> 及 <i>Natural_cloud(white)</i> 所使用的區域(綠色區域)	43
圖 54 : (a) <i>Tile</i> ;(b) <i>Water waves</i> ;	43
圖 55 : 背景紋理實驗的 Rate distortion curves	43
圖 56: <i>Tile</i> 在 Qp37 之主觀視覺(a) Original;(b) HM+BCIM: 0.42 bits/pixel;(c) HM+LinearPlane (LC): 0.50 bits/pixel;(d) HM+LinearPlane (GT): 0.39 bits/pixel;(e) HM+SADCT (GT): 0.41 bits/pixel;(f) HM+Matching Pursuit (GT): 0.42 bits/pixel;	44
圖 57 : <i>Water waves</i> 在 Qp37 之主觀視覺(a) Original;(b) HM+BCIM: 0.29 bits/pixel;(c) HM+LinearPlane (LC): 0.42 bits/pixel;(d) HM+LinearPlane (GT): 0.32 bits/pixel;(e) HM+SADCT (GT): 0.34 bits/pixel;(f) HM+Matching Pursuit (GT): 0.35 bits/pixel;... ..	45
圖 58 : Linear Consensus (LC)分割出的桃紅色前景與綠色背景(a) <i>Tile</i> ;(b) <i>Water waves</i> ;... ..	45
圖 59 : <i>Tile</i> 在不同雜訊強度的 Rate distortion curves	46
圖 60 : <i>Water waves</i> 在不同雜訊強度的 Rate distortion curves	47
圖 61 : SADCT 中橫向 DCT 的配對	49
圖 62 : DCT 係數其最大相關性之配對	49
圖 63 : MPEG4 壓縮結果(a) Original;(b) SADCT;	49

表目錄

表 1：實驗架構	14
表 2：BCIM 之 bits-rate saving	15
表 3：使用預測(Prediction)及無預測(No Prediction)時 BCIM 之 bits-rate saving	15
表 4：圖 24(a)中左上角 16x16 的原始像素值(Original)，紅色部分為原本白色背景 ...	16
表 5：圖 24(b)中左上角 16x16 的預測值(Prediction)，灰色部分為 DC smooth 之預測值	16
表 6：圖 24(c)中左上角 16x16 的重建值(Reconstruction)，紅色部分為原本白色背景	16
表 7：圖 24(a)中左上角 16x16 的餘值(Residual)，紅色部分為原本白色背景	17
表 8：圖 24(a)中左上角 16x16 經由 BCIM 量化後的餘值(Quantize residual)，紅色部分為原本白色背景	17
表 9：圖 25 之區塊且 $Y=\{1, 2\}$ 之情形	18
表 10：實驗架構	29
表 11：前景 BCIM、背景漸層平面、切割線性切割(中位數)之 bits-rate saving	30
表 12：前景 BCIM、背景漸層平面、切割線性切割(平均數)之 bits-rate saving	30
表 13：BCIM 之 bits-rate saving	30
表 14：前景 BCIM、背景漸層平面、切割線性切割(中位數)之 bits-rate saving	32
表 15：前景 BCIM、背景漸層平面、切割線性切割(平均數)之 bits-rate saving	32
表 16：BCIM 之 bits-rate saving	33
表 17：實驗方法說明	34
表 18：前景 SADCT、背景漸層平面、切割線性切割(平均數)之 bits-rate saving	37
表 19：前景 SADCT、背景漸層平面、切割線性切割(平均數)之 bits-rate saving	37
表 20：前景 BCIM、背景 SADCT、切割線性切割(平均數)之 bits-rate saving	38
表 21：前景 BCIM、背景 Matching Pursuit、切割線性切割(平均數)之 bits-rate saving...	38
表 22：實驗方法說明	39
表 23：Matching Pursuit 於 <i>Natural_cloud(black)</i> 及 <i>Natural_cloud(white)</i> 所占的面積 ...	42
表 24：SADCT 於 <i>Natural_cloud(black)</i> 及 <i>Natural_cloud(white)</i> 所占的面積	42
表 25：BCIM 於 <i>Natural_cloud(black)</i> 及 <i>Natural_cloud(white)</i> 所占的面積	42

第一章、緒論

1.1 背景介紹

由於數位內容的成長，因此每天都有許多的人工影像產生，這些由電腦所產生的人工影像，包括像是投影片播放、電視跑馬燈及電玩遊戲進行畫面等，而這類影像大都是從電腦螢幕畫面所擷取出來的，所以統稱為視窗影像(Screen Content Image and Sequence)。另外由於近年來雲端服務、無線傳輸及電玩遊戲實況等應用日益受到重視，因此視窗影像的傳輸也顯得日漸重要。然而傳統的影像壓縮技術標準如 H.264、JPEG 等，是針對自然影像的特性做最佳的壓縮處理，且視窗影像的畫面特性是不同與以往自然影像，因此如果使用傳統的編碼技術於視窗影像，其壓縮效能就相對較差，所以能有效的壓縮這類視窗影像，是之後影像壓縮中重要課題。另外針對這類視窗影像所設計的編碼技術，稱之為視窗影像編碼(Screen Content Coding, SCC)。

不同於以往的自然影像，視窗影像經常包含著許多高對比且顏色單調的影像內容，例如文字及電腦圖形等，由於高對比的特性，使得像素間的相關性不如自然影像般，因此傳統以 Discrete cosine transform (DCT)為主的編碼方式，其視窗影像經過轉換後，並沒有造成能量集中於低頻現象，反而存在著較多的高頻訊號成分，因而降低了壓縮上效率。

為了因應未來應用上的需求，在最新的高效能視訊壓縮編碼(High Efficiency Video Coding, HEVC)會議中，也為 SCC 的制定設立了一討論群，並嘗試加入了些簡單的 SCC 演算法，並在不增加過多的負擔下，能有效提升視窗影像的壓縮效能。



圖 1：Class F 視窗測試影像(a) BasketballDrillText;(b) ChinaSpeed;(c) SlideEditing;(d) SlideShow;

1.2 研究貢獻

視窗影像編碼通常根據其影像特性做設計，因此將針對如圖 2(a)較複雜文字覆蓋於一漸層背景時，以及圖 2(b)當文字覆蓋於自然圖像上時，本篇論文首先將區塊內像素分成文字像素(Text)及圖像像素(Pictorial pixels)兩種，其中圖像像素為非文字特性之像素，例如漸層背景或自然影像背景，另外由於現有 BCIM 架構對於文字像素能有效表示，並有同時表達多個不同平面(平面為單一顏色)的概念，但是並無法同時表示過多的顏色，因此本文修改現有 BCIM 的架構，使之能有效表示文字像素，並可以用一個特殊平面來表示圖像像素，而針對圖像像素則會依據不同特性使用 2 種不同的平面表示，以下將分別介紹。

1. 確定性關係的平面演算法 (deterministic relation algorithm)

首先我們發現有些圖像像素(Pictorial pixels)其呈現一種特定的確定性關係，例如漸層的背景平面，由於肯定其像素間的關係，為了能有效表達這類像素，本篇論文使用已經確定的方程式來表示此一關係，例如本文所使用的漸進方程式來表示其漸層關係。另外由於使用線性方程式表示，因此我們也利用方程式變數有限的特性，發展切割線性平面演算法

2. 統計上相關的平面演算法 (statistical relation algorithm)

由於並非所有的圖像像素(Pictorial pixels)均呈現確定性關係，例如自然圖像，為了也能夠有效表示這些像素，本方法使用統計上之關係來決定最佳的方程式表示，而統計上的關係大都採用 DCT 方式表示，因此本論文也使用 DCT 的方式做運算，由於圖像像素並非完整的區塊，因此需額外處理，而在 MPEG4 中有使用填補空缺(Hole filling)或是採用直接執行 DCT 於不規則區塊的 Shape-adaptive DCT (SADCT)，根據[1]所示 SADCT 的效能優於填補空缺(Hole filling)方法，因此本文將採用 SADCT 來表示有空缺之自然圖像像素。

從實驗中可以發現當使用線性平面時，在客觀上的結果和 BCIM 差不多，然而當像素間存在漸層關係時則可提供較佳的主觀視覺效果，另外提出了的分割方法，能分割出其漸層背景。最後由於 SADCT 在主觀有些問題，且客觀數據上不如 Matching pursuit[2](不須補洞也可執行 DCT 的方法)，因此最後將討論 SADCT 無法有效表示之原因。

1.3 論文編排

本篇論文的編排如下：第二章為現今 SCC 演算法的簡介；第三章為介紹本篇所提出的特殊平面演算法及架構；第四章為本篇方法的實驗結果，最後討論 SADCT 的問題；第五章為結論。

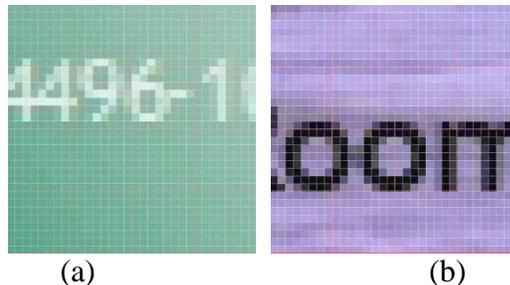


圖 2：視窗影像(a) 漸層背景;(b) 自然影像背景;

第二章、 相關研究

由於視窗影像的應用日益增加，例如遠端桌面、無線傳輸等，且此類影像的特性並不適用於傳統 DCT 的壓縮方法，因此這幾年有許多人開始研究；並依此類影像特性設計壓縮演算法，而在眾多編碼方法中，除了單純取消 DCT 的 Transform skip (TS 編碼方式外)[3][4]，其餘大致上可依不同的編碼方法，分類以下幾類；基於形狀之編碼方式、基於調色盤之編碼方式、區域內顏色之編碼方式、分層架構的編碼方式、基於字典架構之編碼方式。以下將分別介紹這幾類的編碼方式。

2.1 基於形狀之編碼方式 (Shape based)

由於以往的電腦視窗影像，其像素間顏色較單調且連續，因此根據視窗桌面影像的特性採用一種形狀的編碼方式，例如遠端桌面連線的 Virtual Network Computing (VNC)協定[5]中一部分的編碼方式，其編碼方式如圖 4 所示，為將視窗影像切割成許多單一顏色的小塊方形組成，並標示每個方形的座標及顏色，由於視窗影像含有連續顏色的區塊特性，此類方法能用座標及顏色值來有效表示多個相同顏色的像素。

而其分割相同顏色之區塊方法，如圖 5 所示，由左下角之像素出發，畫出最長相同顏色之矩形 (藍色 2 號，8 個像素)及最寬相同顏色之矩形 (紅色 1 號，9 個像素)，之後選擇最多像素之矩形唯一確定切出之區塊 (此時為紅色 1 號)，之後接續以上步驟直到切出所有相同顏色之區塊。

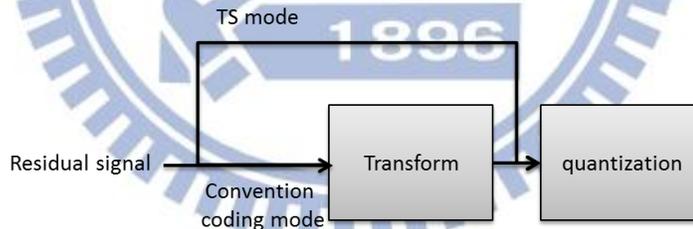


圖 3：TS 流程圖

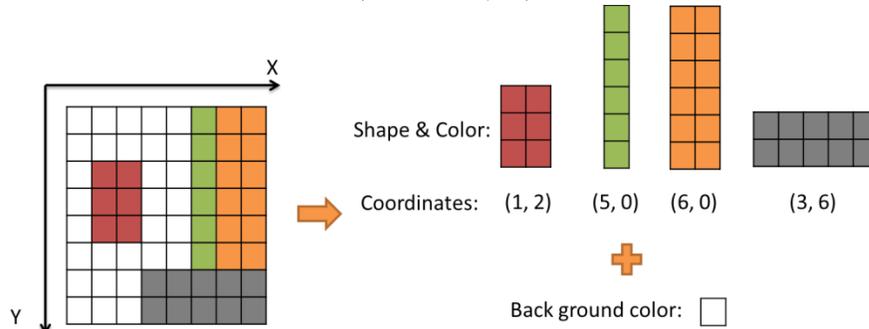


圖 4：根據形狀編碼之架構

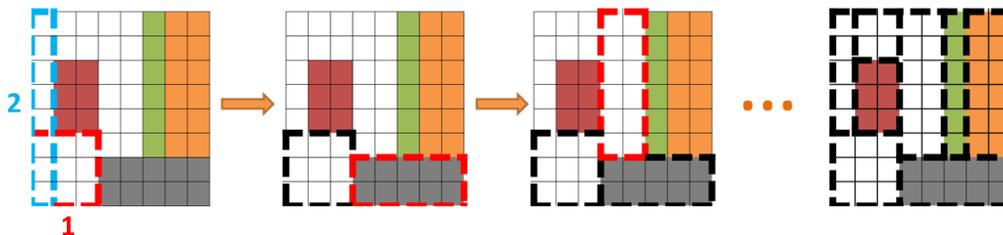


圖 5：切割方塊流程圖

2.2 基於調色盤之編碼方式 (Palette based)

視窗影像中的顏色數量通常較為稀少，但有時在一區塊內，同一顏色並不一定會連續出現，而是分散在區塊內，所以無法使用形狀樣式的編碼方式，因此 VNC 另一部份編碼概念提出了針對區塊內顏色數量稀少的調色盤式設計，此方法為傳送區塊內的所有顏色值，來減少標示每像素值所需花費的位元數，如原本 8x8 大小的區塊內只包含 4 種顏色，且每個顏色花費為 8 bits，則此方法只需傳送 4x8 bits 的顏色位元，以及 8x8x2 bits 來標示所有像素值，相較於原本 8x8x8 bits 的原始資料可達到一定壓縮效果。

在調色盤的編碼方式中，其原始架構為不失真的壓縮方法，然而視窗影像中有些顏色是不為人眼所察覺，因此為了能達到更好的壓縮效果，[6][7][8][9]採取了有量化的編碼方式，稱之為 Base Color Index Map (BCIM)。BCIM 是一種搭配非均勻量化 (Non-uniform quantization) 的演算法，如所示會將輸入的區塊內像素量化成幾種重建值 (Base color)，並利用索引圖 (Index map) 的方式重建出原有的區塊，非均勻量化相較於原本的均勻量化演算法，由於需額外傳送重建值，因此相較於均勻量化會有多的額外開銷 (Overhead)，但是由於考慮了所有區塊內的像素資訊進行量化，因此在不考慮額外開銷的情況下，BCIM 可在固定重建值個數下，達到最少的量化誤差。

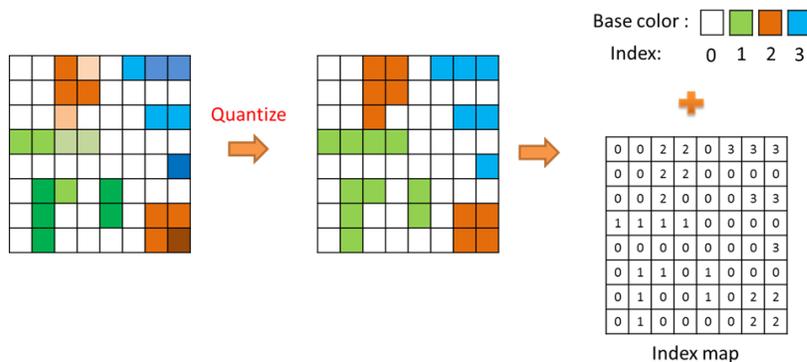


圖 6：BCIM 之架構

LeftUp Pixel	Up Pixel	RightUp Pixel
Left Pixel	Current Pixel	

圖 7：Context model 所參考的像素位置

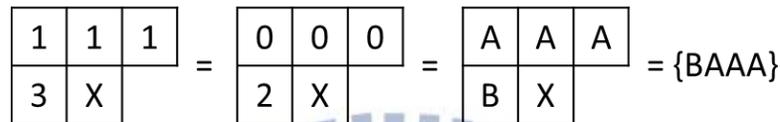


圖 8：Context model 的選擇方法

$$f(x) = \begin{cases} 0, & x = A \\ 1, & x = B \\ 2, & x = C \\ 3, & x = D \end{cases}$$

圖 9：調整索引值的方法

從圖 6 的架構中可以知道，BCIM 所需編碼的資訊為重建值個數、重建值及索引圖，而實作中重建值個數及重建值採用固定長度編碼方式，索引圖則使用光柵搜尋(Raster scanned)的方式，並配合 CABAC 的 Context model 做編碼。目前壓縮索引值的 Context model 會依據其左邊、左上、上及右上的 4 個索引值如圖 7 所示做選擇。其判斷會根據不同索引值間的排列組合，而非參考其數值，如圖 8 所示當壓縮 X 位置的索引值時，其左上、上及右上的值相同，而僅左邊值不相同，則歸類為 {BAAA} 這類型的排列，並非參考其各別數值為何，因此共有 15 種的組合 {AAAA, AAAB, AABA, ABAA, BAAA, AABB, ABAB, ABBA, AABC, BAAC, BCAA, ABAC, ABCA, BACA, ABCD}，也就是有 15 組 Context model，根據其 Context model 的選擇最後做調整索引值的動作，根據圖 9 的式子，當目前壓縮 X 位置的索引值和 A 的索引值相同時，則調整索引值為 0，如果和 B 相同則調整索引值為 1，依此類推，其目的為統一所有 Context model 的設計，並且也可依據重新調整後數值較小的索引值，給予花費較少的代價。例如圖 8 中最左邊 {3111} 的例子，其對應的 Context model 為 {BAAA}，當 X=1 時根據圖 9 則調整後的索引值為 0。

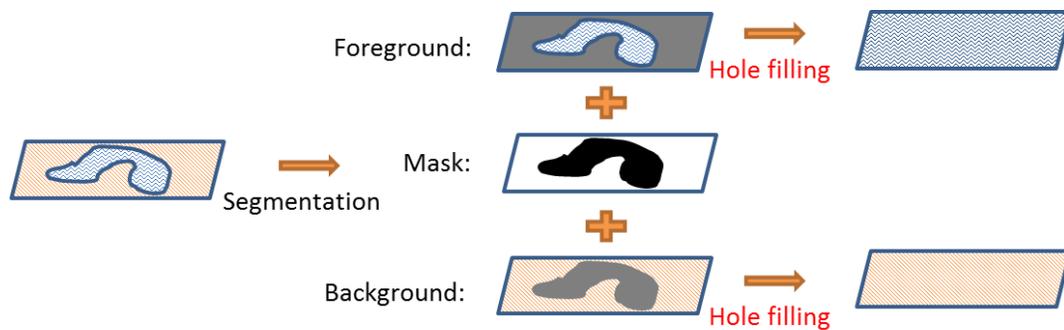


圖 10：MRC 架構圖

2.3 分層架構之編碼方式 (Layer based)

由於在視窗影像中，會有完全不同特性的影像疊加在一起，例如文字疊加於自然影像上，因此為了使相同特性的影像能一起壓縮，因此有了分層架構的編碼方式，其架構為Mixed Raster Content (MRC)影像分層的模型，[10][11]說明了MRC的基礎架構，MRC的方法主要為利用多層影像並搭配遮罩的方式來表示原始的單一影像，基本的架構可將原始影像切割成三層圖層，分別為前景、背景及二元遮罩層，利用切割演算法分層，之後再對各圖層採用不同的壓縮演算法，如圖10所示。

二元遮罩層的功能為分別表示原始影像中前景及背景的部分，通常使用二元影像編碼技術壓縮，如JBIG和JBIG2。而前景和背景圖層部分，則使用JPEG壓縮演算法進行前景及背景的編碼，另外[12][13]則在MRC的架構修改了其前景及背景圖層的編碼演算法，[12]採用IW44的演算法，[13]則使用了H.264中的Intra coding。另外[11]中有對MRC的架構做更詳細的介紹，並提出在MRC架構下相較於傳統影像編碼的好處。在切割分層的演算法中，[14]提出了一種利用閾值做二元切割的最佳演算法，此演算法為比較閾值和像素間大小作為判斷分割的依據，例如比閾值大為前景，閾值小為背景，而閾值則是使用Lagrange cost function去比較不同閾值的切割中何者為最佳值，以便做最佳切割。

2.3.1 填補空缺之演算法 (Hole filling)

由於分層後其前景及背景圖層分別會產生影像空缺的現象，這現象我們稱之為洞，另外因為洞的關係，所以無法使用 DCT 進行轉換，因此在編碼前會先將洞填滿，以下將介紹填補空缺之演算法，[12][15][16]分別提出了使用遞迴方式填補洞的演算法

◇ 參考鄰近資料的填補方法

[15]由於鄰近的像素間通常包含定的關聯性，因此將參考空缺鄰近的資料，填補空缺的資料，如圖 11 所示 U 為存在的像素，X 為空缺的像素，當 X 附近有 U 的像素時(如藍色 X)，則將以 X 為中心，其上、下、左、右四個存在的 U 取平均作為填補 X 的數值，X 附近若全無 U 則跳過等待一下遞迴，遞迴執行直到所有 X 填補完成。

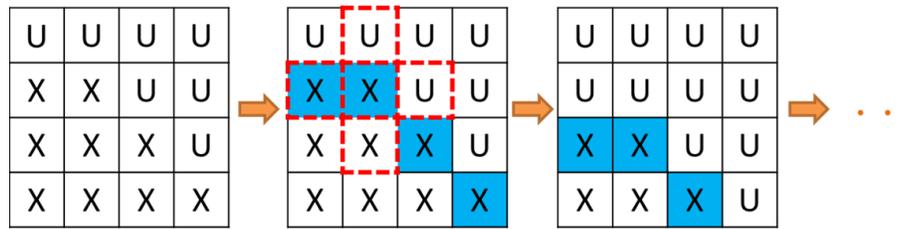


圖 11：參考鄰近資料的填補方法

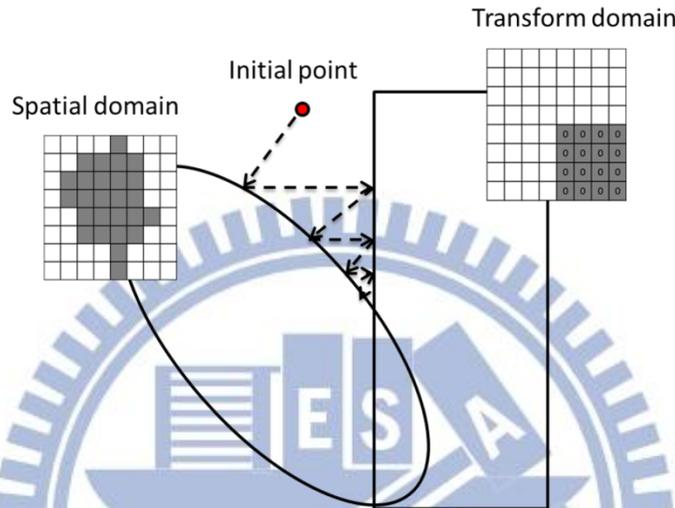


圖 12：POCS 概念圖

◇ 利用 Projections onto Convex Sets (POCS)的填補演算法

此方法[12]的目標為填補完的區塊，進行DCT轉換後其部分的AC值為0，並採用POCS的方法求解，此演算法為設定兩個區塊集合如圖12所示，第一個集合為區塊內的空缺(灰色部分)任意填補所生成之區塊集合，而第二個集合為區塊經過DCT轉換後特定AC值(灰色部分)為0之區塊集合，此時由於此兩個集合均為凸集，當此兩集合存在交集時，根據POCS的概念，可重複投影至這兩集合，並且最後會收斂且投影至交集部分。

2.3.2 Matching pursuit直接計算DCT係數

[2]所提出的方法，並不採取補洞的方式，而是使用Matching pursuit的演算法直接求出在頻率域上的值，以下將介紹Matching pursuit的編碼方式。這是一種類似DCT的轉換法，而DCT的基本概念如式子(1)所示為使用一組n維的DCT基底 $(\vec{b}_1 \cdots \vec{b}_n)$ ，並計算出各基底所對應的係數 $(c_1 \cdots c_n)$ 來表示此區塊內的n個像素 (\vec{P}) ，然而由於特殊平面並不是完整n維的區塊，但是又不改變DCT基底在空間域空間中相對應的位置情況底下，因此便將基底對應的像素空缺部分設為0，再計算其係數，如式子(2)所示假設 p_2 位置的像素為空，則將所對應的 $b_{12} \cdots b_{n2}$ 設為0再求其係數 $(c_1 \cdots c_n)$ ，然而由於特殊平面的像素個數維度小於n，卻

有 n 個基底求解，因此這是一個 over complete function 會有無限多組解，而為了能儘量達到能量集中的效果，所以將採用 Matching pursuit 的演算法求解。

$$\vec{p} = c_1 \vec{b}_1 + c_2 \vec{b}_2 + c_3 \vec{b}_3 + \dots + c_n \vec{b}_n \quad (1)$$

$$\begin{pmatrix} p_1 \\ 0 \\ p_3 \\ \vdots \\ p_n \end{pmatrix} = c_1 \begin{pmatrix} b_{11} \\ 0 \\ b_{13} \\ \vdots \\ b_{1n} \end{pmatrix} + c_2 \begin{pmatrix} b_{21} \\ 0 \\ b_{23} \\ \vdots \\ b_{2n} \end{pmatrix} + c_3 \begin{pmatrix} b_{31} \\ 0 \\ b_{33} \\ \vdots \\ b_{3n} \end{pmatrix} + \dots + c_n \begin{pmatrix} b_{n1} \\ 0 \\ b_{n3} \\ \vdots \\ b_{nn} \end{pmatrix} \quad (2)$$

Matching pursuit 是一種利用數值方法來求解的演算法，其可以從 over complete function 的解集中求出一組解，主要概念為將原始向量不斷投影至最佳的基底上並求出其對應的係數，直到收斂，而基底可從 over complete dictionary 中選擇，而在目前的例子中，令式子 (2) 中的 $(b_{11} \ 0 \ b_{13} \ \dots \ b_{1n})^T = \vec{b}'_1$ 及 $(p_1 \ 0 \ p_3 \ \dots \ p_n)^T = \vec{p}'$ ，則此 over complete dictionary (D) 為 $\vec{b}'_1, \vec{b}'_2, \vec{b}'_3, \dots, \vec{b}'_n$ 的集合，如此可表示成式子 (3)， \vec{p}' 為原始的向量，其中 t 為投影的次數， a_t 為每次投影在 \vec{b}'_{nt} 上的係數，也就是從 over complete dictionary (D) 中所選擇最佳 n 的基底 \vec{b}'_{nt} 及其對應係數 a_t ，最後將相同 n 之基底 \vec{b}'_{nt} 所對應之 a_t 相加，則可求得 \vec{b}'_n 所對應的係數 c_n 。

$$\vec{p}' = \sum_{t=1}^{+\infty} a_t \vec{b}'_{nt}, \quad \vec{b}'_{nt} \in D \quad (3)$$

上述說明了 Matching pursuit 的基本概念，而其演算法採用遞迴的方式進行，如圖 13 所示，首先 \vec{p}' 為原始的向量並設定為 \vec{r}_1 ，將其投影至 D 中最佳的基底 \vec{b}'_{n1} ，其中最佳的基底是和 \vec{r}_1 做內積，其內積絕對值最大的基底，再算出其對應的係數 a_1 ，之後再將 \vec{r}_1 扣除投影在基底 \vec{b}'_{n1} 上的向量得到 \vec{r}_2 ，在重複以上動作，對 \vec{r}_2 尋找最佳的基底 \vec{b}'_{n2} 及其對應係數 a_2 ，並扣除 \vec{r}_2 投影在基底 \vec{b}'_{n2} 的向量得到 \vec{r}_3 ，不斷重複，直到 \vec{r}_t 的長度小於一個閾值則演算法結束，如此即可求得每次 t 投影所對應的最佳 n 的基底 \vec{b}'_{nt} 及其對應係數 a_t 。

Input : signal: \vec{p}' , dictionary : D , $\vec{b}'_{nt} \in D$
 Output : list of coefficients : (a_t, \vec{b}'_{nt})
 Initialization :
 $\vec{r}_1 = \vec{p}'$
 $t = 1$
 Repeat :
 find $\vec{b}'_{nt} \in D$ with maximum inner product $|\langle \vec{r}_t, \vec{b}'_{nt} \rangle|$
 $a_t = \langle \vec{r}_t, \vec{b}'_{nt} \rangle$
 $\vec{r}_{t+1} = \vec{r}_t - a_t \vec{b}'_{nt}$
 $t = t + 1$
 Until stop condition : $\|\vec{r}_t\| < threshold$

圖 13：Matching pursuit 演算法

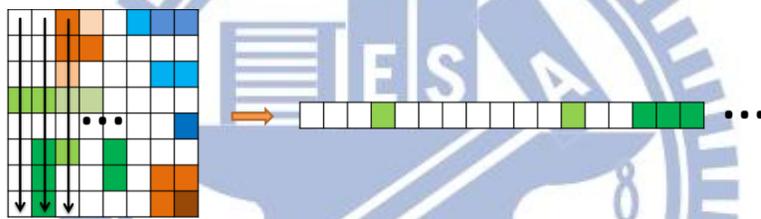


圖 14：將二維視窗影像區塊掃描成一維陣列

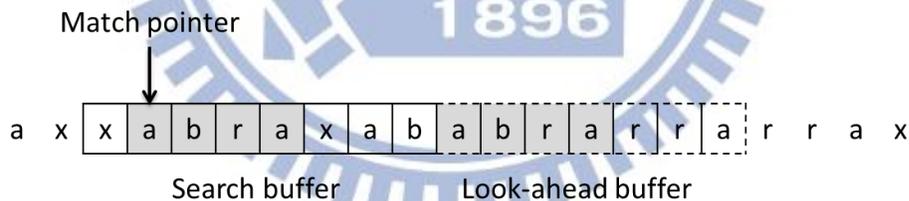


圖 15：字典編碼概念

2.4 基於字典架構之編碼方式 (Dictionary based)

這種架構方法為將二維的視窗影像區塊使用垂直掃描成一維的陣列如圖 14，並將每個像素表示成單一字元，再直接採用傳統字典的編碼方式 [17][18][19]，例如 LZ、Lempel-Ziv-Welch (LZW) 等，而字典編碼的主要精神為將過去的字串資料用字典儲存起來，因此當前的資料如果有跟字典中一樣的字串，則可以用標記的方式表示目前的資料字串，如圖 15 所示，由於視窗影像的不同顏色的變化量較少，因此採用字典編碼方式可有效的應用於視窗影像的編碼設計。

在 LZ77 方法中，字典就是先前已編碼的序列的一部分。編碼器透過如圖 15 所示的一扇滑動窗口(實線方塊部分)檢查輸入序列。窗口包含兩個部分，亦即包含最近編碼序列的搜尋緩衝區(search buffer)，以及包含欲編碼序列下一部份的預覽緩衝區(look-ahead buffer)。在

圖 15 中，搜尋緩衝區包含 8 個字元，預覽緩衝區則包含 7 個字元。為了將預覽緩衝區內的序列編碼，編碼器會在搜尋緩衝區內把一個搜尋指標往回移動，直到遇到預覽緩衝區內第一個匹配。匹配指標(match pointer)與預覽緩衝區的距離稱為偏移植(offset)。然後編碼器會檢查指標位置後面的符號，看看是否與預覽緩衝區中的連續符號匹配。搜尋緩衝區內與預覽緩衝區內連續符號匹配的符號個數(從第一個字元開始)稱為匹配長度。編碼器會尋找搜尋緩衝區內最長的匹配。一但發現最長的匹配，編碼器以三元組 (o,l,c) 將它編碼，其中 o 是偏移植， l 是匹配長度， c 是預覽緩衝區內匹配後面的符號對應編碼字。以圖 15 為例，偏移植 o 等於 7，匹配長度 l 等於 4，且預覽緩衝區內匹配後面的符號是 r 。

2.5 摘要

從[20]中可以發現當區塊呈現某些特性時，基於調色盤之編碼方式的確會比基於形狀之編碼方式之效能還好，例如當相同顏色卻分散或形狀不規則等之情況，反之則相反。另外根據[21]中對不同編碼方式比較中，在不同的測試影像及品質要求下，其效能都不同，例如在高品質時採用基於字典架構之編碼方式會較好，然而在較低品質需求下，基於調色盤之編碼方式的編碼效果較好，因此在設計視窗影像編碼上，應對其不同條件及區塊特性下做編碼設計。因此目前視窗編碼方式均有針對其特殊區塊特性做設計，在我們的觀察下，如圖 16 所示為不同樣式的視窗區塊，基於形狀之編碼方式則適用圖 16(a)的作業視窗的圖標區塊，而調色盤式的編碼方式可適用於圖 16(b)般的文字區塊，另外分層架構的編碼方式較為適用於圖 16(c)單一顏色文字且背景為圖像的區塊，最後基於字典架構之編碼方式適用於圖 16(d)般簡單且一直重複相同訊號的區塊，因此視窗影像應針對其不同特性做編碼調整。

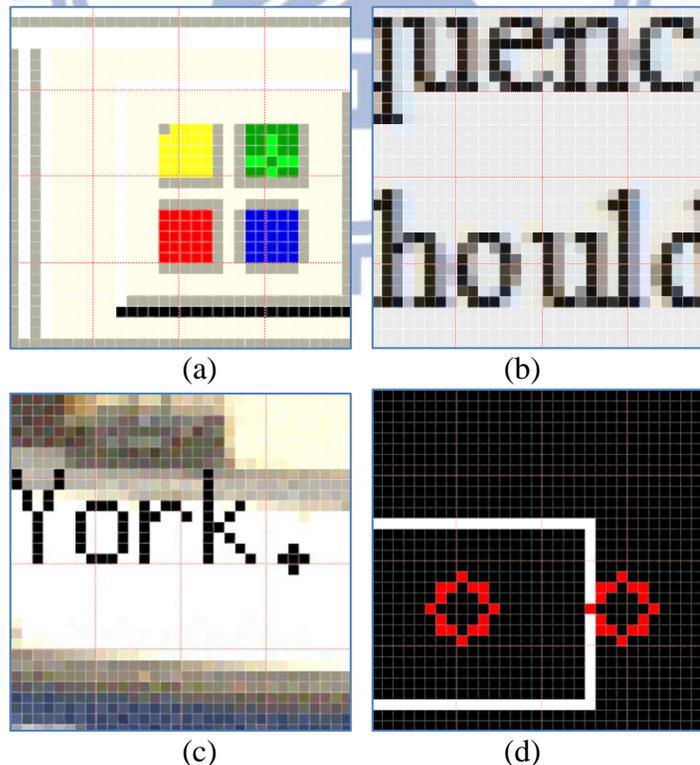


圖 16：視窗影像區塊(a) 作業視窗的圖標;(b) 較複雜的文字;(c) 圖像複合簡單文字;(d) 電子線路圖;

第三章、研究方法

上一章介紹中可以發現，視窗編碼均是針對其影像特性做設計，然而還有些視窗影像的特性並不適用以上方法，如圖 2(a)所示當較複雜的文字覆蓋於一漸層背景時，以及圖 2(b)當文字覆蓋於自然圖像上時，採用傳統 BCIM 等方式可能較無效率。

因此本篇將修改 BCIM，並結合分層的架構，使之能處理更多不同特性的文字區塊。而在分層架構中，本論文將分成兩種不同特性的平面；確定性關係及統計上關係的平面表示方式，然而在統計上關係的表示方法中會利用 DCT 表示其相關性，但由於會有空洞的問題，從相關工作中有使用 hole filling 及 Matching pursuit 等方法處理此問題，但 hole filling 及 Matching pursuit 都會導致資料量上升，另外 Matching pursuit 也有運算時間過久之問題，因此本篇將使用 SADCT 的方式來表示以避免資料量上升。

另外由於本方法會捨棄傳統編碼架構中的 Transform 設計，並採用類似 BCIM 的設計架構，因此以下首先將分析視窗影像的特性及 Transform 所造成的影響，之後使用 BCIM 取代傳統 DCT 架構式影像編碼方法，及分析在 BCIM 的編碼方法下 Intra prediction 的影響。

3.1 視窗影像特性及 BCIM 之分析

本節內容將分析視窗影像的特性及 Transform 所造成的影響，之後使用 BCIM 取代傳統 DCT 架構式影像編碼方法，及分析在 BCIM 的編碼方法下 Intra prediction 的影響。

3.1.1 Transform 對視窗影像影響之分析

在這小節中將分析視窗影像的特性，及在傳統 Hybrid coding 中 Transform 可能帶來的問題，從分析中可以發現 Transform 確實會影響視窗影像，不只無法有 Energy compaction 效果，還會導致其能量分散，另外從 Correlation coefficient matrix 的統計中，視窗影像其像素餘值 (Residual) 與餘值 (Residual) 間的相關性 (Correlation) 很低，因此無法有效的利用 Transform 來表示。

首先我們觀察視窗影像中那些區塊為不同於自然影像特性，而是屬於 Screen content 特性的區域，如圖 17、圖 18 所示，綠色區塊部分為 Screen content 特性的區域，從圖中可以發現，綠色部分大都為邊界交接處及文字部分，如圖 17 紅色圈中天空和背景的交接處，左下方文字的部分，還有圖 18 中其文字區域及電腦生成的圖形邊界等，從以上的觀察，我們可發現，Screen content 特性的區域都含有高對比的影像特性。

針對影像特性的觀察中，可發現其同一區塊內會有類似邊界般像素值不連續的現象，因此我們假設 DCT 並不適用於這類的影像區塊的現象，是因為不連續的像素值可能會造成頻率域上發生能量分散的情況，基於以上假設，我們分別觀察了 Screen content 特性的區塊及自然影像特性的區塊，經由預測所產生的餘值 (Residual)，其空間域及頻率域上量化後的 Index 分佈如圖 19、圖 20 所示，由圖 19 中可以看到自然影像特性的區域，經過 DCT 後接

近 Zero index 的機率上升，表示能量會集中於少數維度上，使得不含能量的維度增加，也就是發生所謂的 Energy compaction 效應，然而在圖 20 中 Screen content 特性的區域卻呈現相反的結果，從圖中可以看到經過 DCT 後，其接近 Zero index 的機率反而下降了，也就表示 DCT 前的能量是更為集中的，然而經過 DCT 後卻造成了能量分散的情形，這也就說明了不做 DCT 能為視窗影像區塊帶來好處的原因。



圖 17：綠色標示為 Screen content 特性的區域(Class F – China Speed)

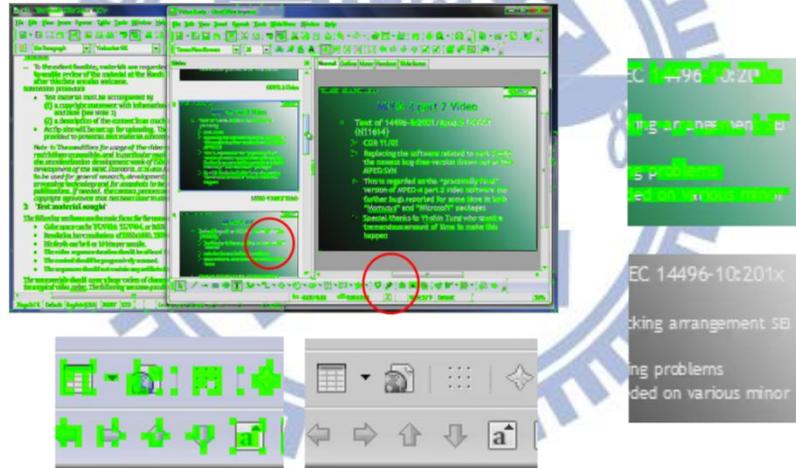


圖 18：綠色標示為 Screen content 特性的區域(Class F – Slide Editing)

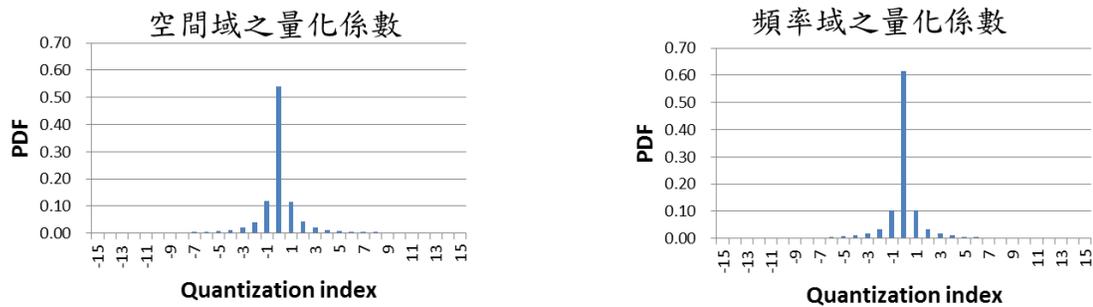


圖 19：自然影像特性的區域 Transform 前後其量化後的 Index 分佈

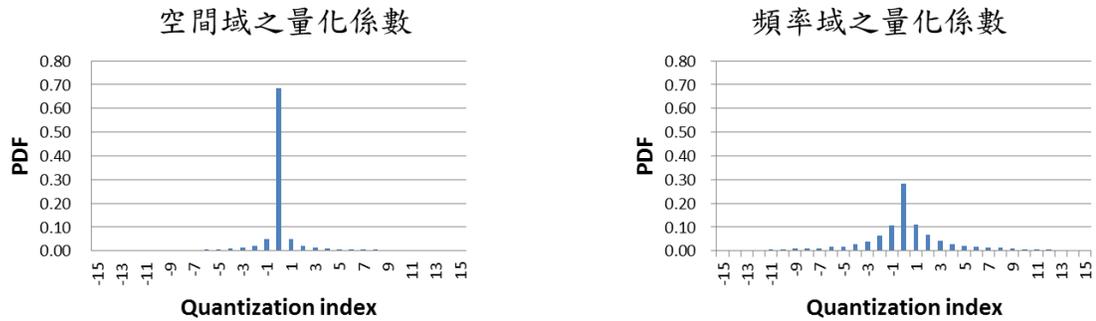


圖 20：Screen content 特性的區域 Transform 前後其量化後的 Index 分佈

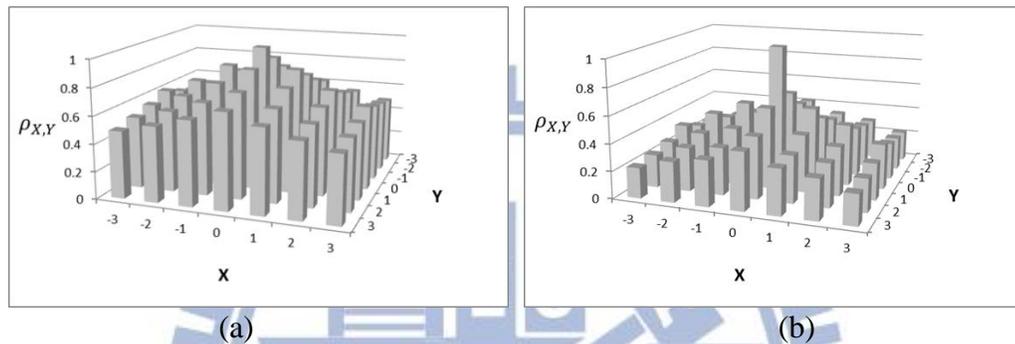


圖 21：原始像素值(Original)間的 Correlation coefficient matrix(a) Transform block;(b) Screen content block;

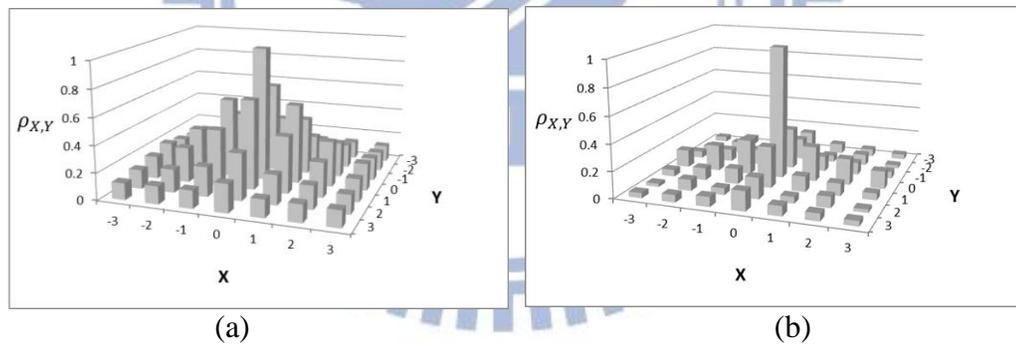


圖 22：餘值(Residual)間的 Correlation coefficient matrix(a) Transform block;(b) Screen content block;

另外針對這些區塊，我們也統計了其像素值及餘值(Residual)的 Correlation coefficient matrix 如圖 21 及圖 22 所示，其中X、Y為像素間水平及垂直距離， $\rho_{x,y}$ 為像素間之相關性(Correlation)，從中可以發現自然影像特性的區塊，其像素餘值(Residual)間都還保有一定的相關性，因此可以使用 Transform 的轉換去除其相關性並達到 Energy compaction 的效果，然而在 Screen content 特性的區塊，其像素間餘值(Residual)幾乎不含有任何相關性，也因此我們無法以統計的方法找出最佳的一組基底，並用來表示此類的區塊，也就代表著 Transform 的方法並不能對此類區塊帶來好處。

3.1.2 使用K-means & BCIM取代量化及熵編碼設計

前一節中我們取消了 Transform 並分析其好處的原因，但由於取消了 Transform，使之後的壓縮過程中不考慮區塊內像素間關係，因此在這情況下只剩下預測工具(Prediction)能減少像素間關係，如圖 21 和圖 22 之比較，而由於像素間距離越近其相關性越高，因此在越小的區塊下其預測(Prediction)效果會較佳，另外從圖 22 的 Correlation coefficient matrix 中可以發現，此類區塊並不適用於傳統的 Transform 並再做最佳量化位元分配的方法，因此為了能做區塊內最佳的量化位元分配來達到更好的壓縮效率，這邊修改了原本的均勻量化，而採用單一區塊內做 k-means 演算法，找出餘值(Residual)的最佳量化位元分配，並可配合 Base Color Index Map (BCIM)這種編碼方式表達量化後的區塊，這裡統稱為 BCIM mode。

實驗結果

實驗平台為 FM7.0，並將 HM7.0 中 Transform skip mode 改成 BCIM mode，比較對象為 HM7.0 使用通用的測試設定，其架構如表 1 所示。

表 1：實驗架構

Reference software	HM7.0
Quantization parameters	22, 27, 32, 37
Coding structure	Intra_HE10(All intra)
CU size	8x8~64x64
PU partition	Symmetric
TU partition	RQT
Intra prediction	DC, Planar, 33 Directions, DM, LM Chroma
Transform	DCT 4x4~32x32, Transform Skip 4x4
In-loop filter	De-blocking, SAO, ALF
Entropy coding	CABAC
Test sequence	Class F

實驗結果如表 2 所示，其中 TU Size 表示為其限制大小，4x4 為限定在 4x4，32x32~4x4 為可動態選擇 32x32~4x4 之間的大小，從表中可以看到，使用 BCIM 在動態選擇區塊大小情況底下位元率可減少約 11%，然而只限定 4x4 區塊的實驗顯示，其所需花費的位元數相較於原本的均勻量化是上升，原因是因為相較於大區塊，小區塊所需的額外消費(Overhead)是較多的，因此准許大區塊情況底下使用 BCIM 會較有好處。

3.1.3 分析 Intra prediction 對 BCIM 之影響

在傳統的編碼架構中，預測(Prediction)是重要的壓縮工具，但是在原始的 BCIM 設計中並沒有使用預測工具，也就是直接對原始影像作 BCIM 的量化，而不是對預測後的餘值(Residual)做量化，因此本節將比較使用預測(Prediction)和沒有預測(No prediction)時對 BCIM 的影響。

實驗結果

實驗平台為 FM7.0，並將 HM7.0 中 Transform skip mode 改成 BCIM mode，比較對象為 HM7.0 使用通用的測試設定，其架構如表 1 所示。從實驗結果表 3 中可看到無預測(No prediction)時 BCIM 的位元率較預測時減少了約 2%，且在主觀視覺上，如圖 23 紅色圈所示 BCIM 使用預測時會造成主觀上有些特殊的失真。

表 2：BCIM 之 bits-rate saving

	Y	U	V
4x4	4.0%	2.8%	3.1%
32x32~4x4	-11.4%	-8.9%	-8.3%

表 3：使用預測(Prediction)及無預測(No Prediction)時 BCIM 之 bits-rate saving

	Y	U	V
Prediction	-11.4%	-8.9%	-8.3%
No Prediction	-13.6%	-10.3%	-9.7%

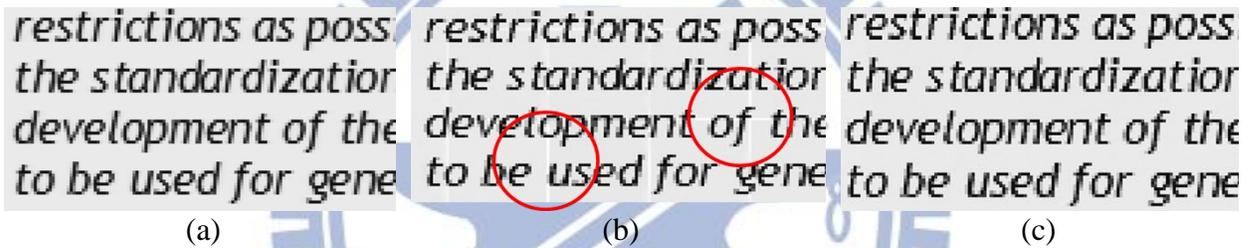


圖 23：有無預測(Prediction)時 BCIM mode 之主觀差異(a) Original;(b)BCIM with prediction;(c) BCIM without prediction;

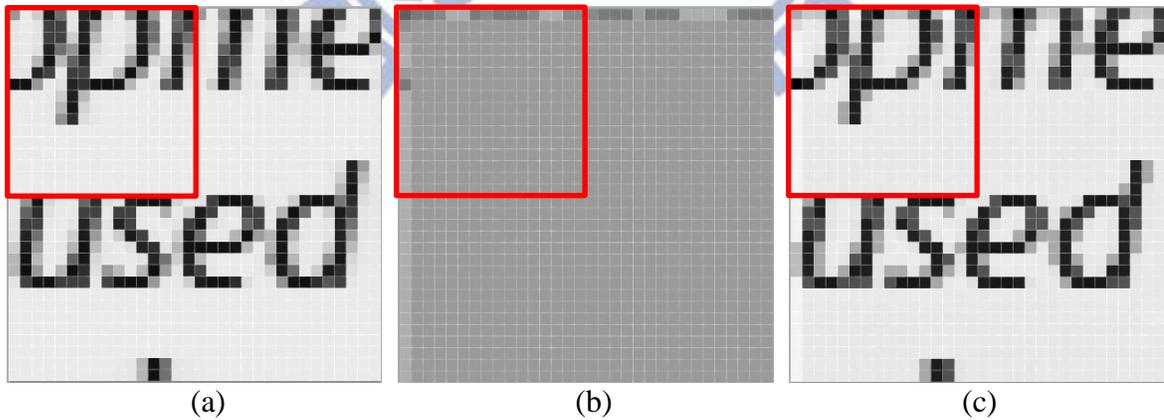


圖 24：區塊內像素(a)原始像素值(Original);(b)預測值(DC prediction);(c)重建值

表 4：圖 24(a)中左上角 16x16 的原始像素值(Original)，紅色部分為原本白色背景

608	932	820	108	440	912	144	92	844	932	584	64	608	932	404	64
916	936	908	248	256	912	140	624	916	932	892	140	448	936	396	264
920	932	908	248	256	820	128	744	920	932	892	140	448	924	256	448
920	928	892	136	452	588	88	856	920	928	820	104	612	912	140	624
920	928	728	80	740	388	180	896	928	928	588	88	852	820	128	744
920	912	144	264	924	248	260	932	920	724	64	616	936	724	88	856
64	72	608	928	892	140	204	68	64	84	736	928	920	392	180	900
936	936	936	928	820	124	740	924	936	936	936	936	936	936	936	936
936	936	936	924	728	88	844	920	936	936	936	936	936	936	936	936
936	936	936	928	584	180	884	920	936	936	936	936	936	936	936	936
936	936	936	936	936	936	936	936	936	936	936	936	936	936	936	936
936	936	936	936	936	936	936	936	936	936	936	936	936	936	936	936
936	936	936	936	936	936	936	936	936	936	936	936	936	936	936	936
936	936	936	936	936	936	936	936	936	936	936	936	936	936	936	936
936	936	936	936	936	936	936	936	936	936	936	936	936	936	936	936
936	936	936	936	936	936	936	936	936	936	936	936	936	936	936	936
936	936	936	936	936	936	936	936	936	936	936	936	936	936	936	936
936	936	936	936	936	936	936	936	936	936	936	936	936	936	936	936

表 5：圖 24(b)中左上角 16x16 的預測值(Prediction)，灰色部分為 DC smooth 之預測值

385	488	488	548	693	693	488	548	488	488	488	548	693	693	548	548
567	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
656	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
612	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
483	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615
694	615	615	615	615	615	615	615	615	615	615	615	615	615	615	615

表 6：圖 24(c)中左上角 16x16 的重建值(Reconstruction)，紅色部分為原本白色背景

695	798	798	42	408	1003	203	42	798	798	554	42	759	1003	263	42
877	925	925	330	330	925	109	681	925	925	925	109	330	925	330	330
966	925	925	330	330	925	109	681	925	925	925	109	330	925	330	330
1004	925	925	109	330	681	109	925	925	925	925	109	681	925	109	681
1004	925	681	109	681	330	109	925	925	925	681	109	925	925	109	681
922	925	109	330	925	330	330	925	925	681	109	681	925	681	109	925
0	109	681	925	925	109	109	109	109	109	681	925	925	330	109	925
1004	925	925	925	925	109	681	925	925	925	925	925	925	925	925	925
1004	925	925	925	681	109	925	925	925	925	925	925	925	925	925	925
1004	925	925	925	681	109	925	925	925	925	925	925	925	925	925	925
1004	925	925	925	925	925	925	925	925	925	925	925	925	925	925	925
1004	925	925	925	925	925	925	925	925	925	925	925	925	925	925	925
1004	925	925	925	925	925	925	925	925	925	925	925	925	925	925	925
1004	925	925	925	925	925	925	925	925	925	925	925	925	925	925	925
1004	925	925	925	925	925	925	925	925	925	925	925	925	925	925	925
1004	925	925	925	925	925	925	925	925	925	925	925	925	925	925	925

表 7：圖 24(a)中左上角 16x16 的餘值(Residual)，紅色部分為原本白色背景

223	444	332	-440	-253	219	-344	-456	356	444	96	-484	-85	239	-144	-484
349	321	293	-367	-359	297	-475	9	301	317	277	-475	-167	321	-219	-351
264	317	293	-367	-359	205	-487	129	305	317	277	-475	-167	309	-359	-167
226	313	277	-479	-163	-27	-527	241	305	313	205	-511	-3	297	-475	9
226	313	113	-535	125	-227	-435	281	313	313	-27	-527	237	205	-487	129
308	297	-471	-351	309	-367	-355	317	305	109	-551	1	321	109	-527	241
-419	-543	-7	313	277	-475	-411	-547	-551	-531	121	313	305	-223	-435	285
242	321	321	313	205	-491	125	309	321	321	321	321	321	321	321	321
242	321	321	309	113	-527	229	305	321	321	321	321	321	321	321	321
242	321	321	313	-31	-435	269	305	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
242	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321

表 8：圖 24(a)中左上角 16x16 經由 BCIM 量化後的餘值(Quantize residual)，紅色部分為原本白色背景

310	310	310	-506	-285	310	-285	-506	310	310	66	-506	66	310	-285	-506
310	310	310	-285	-285	310	-506	66	310	310	310	-506	-285	310	-285	-285
310	310	310	-285	-285	310	-506	66	310	310	310	-506	-285	310	-285	-285
310	310	310	-506	-285	66	-506	310	310	310	310	-506	66	310	-506	66
310	310	66	-506	66	-285	-506	310	310	310	66	-506	310	310	-506	66
310	310	-506	-285	310	-285	-285	310	310	66	-506	66	310	66	-506	310
-506	-506	66	310	310	-506	-506	-506	-506	-506	66	310	310	-285	-506	310
310	310	310	310	310	-506	66	310	310	310	310	310	310	310	310	310
310	310	310	310	310	66	-506	310	310	310	310	310	310	310	310	310
310	310	310	310	310	66	-506	310	310	310	310	310	310	310	310	310
310	310	310	310	310	310	310	310	310	310	310	310	310	310	310	310
310	310	310	310	310	310	310	310	310	310	310	310	310	310	310	310
310	310	310	310	310	310	310	310	310	310	310	310	310	310	310	310
310	310	310	310	310	310	310	310	310	310	310	310	310	310	310	310
310	310	310	310	310	310	310	310	310	310	310	310	310	310	310	310
310	310	310	310	310	310	310	310	310	310	310	310	310	310	310	310

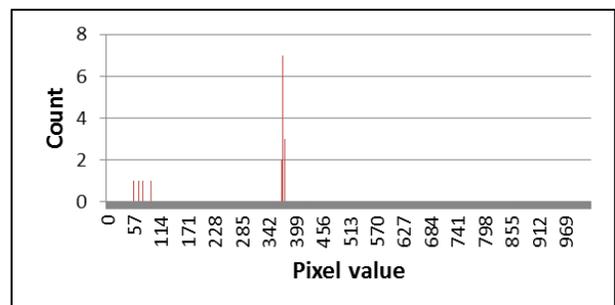
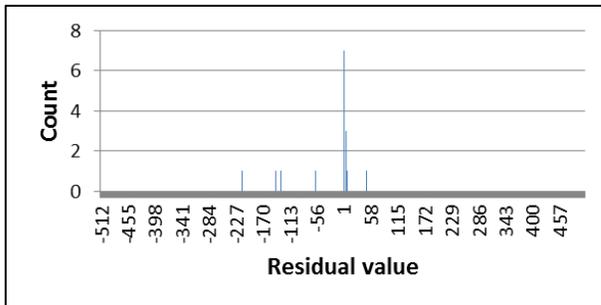


圖 25：有預測及無預測其區塊內像素值的分佈圖

表 9：圖 25 之區塊且 $Y=\{1, 2\}$ 之情形

	Var(X)	E[Var(X Y)]	Var(E[X Y])
No Prediction	16847.75	47.67	16800.08
Prediction	4791.16	615.19	4175.97

由以上實驗可以發現，不管在主觀或客觀上，預測工具是不利於 BCIM 的設計，這在傳統編碼設計概念上是相反的，因此我們將進一步分析預測工具對 BCIM 的影響。

首先在主觀視覺方面，可以發現在圖 23(b)中的重建值(Reconstruction)有不連續的影像結構(紅色圈)，另外從圖 24 及表 4~表 6 可以發現，其原始像素(Original)經由預測並量化重建後，會造成在原始像素為連續之區域(表 4 虛線紅框中數值)，而重建後其重建值(表 6 虛線紅框中數值)卻出現不連續的現象，這是由於在文字區塊中大都並沒有方向性，因此其 Prediction 大都採用 DC prediction(預測值為參考像素的平均)，但由於 HM 中的 DC Prediction 會使用 DC smooth(邊界的預測值會再做一次 filter)，因此其預測值會如表 5 般，鄰近邊界和其他位置之預測值不相同，因此原始像素值相同的像素，經由預測後其餘值(Residual)會如表 7(虛線紅框中數值)般不相同，然而經由 BCIM 量化後原本不相同之數值量化成相同數值(表 8 虛線紅框中數值)，因此重建時搭配表 5(虛線紅框中數值)的預測值，便會重建出不連續之現象。

在客觀數據上如表 3 所示，無預測工具比使用預測工具表現較好，因此我們將觀察在 Screen content 的這些區塊下，無預測(No prediction)和使用預測(Prediction)下其進行 BCIM 量化後的誤差量及所需位元的影響，首先我們先觀察在單一區塊內的變化，其中圖 25 為此區塊有預測及無預測的分布，並分別計算其變異量(Variance)，這裡我們使用 Law of total variance 計算其變異量如式子(4)所示，其中 X 為區塊內個別像素值(Residual 或 Pixel value)，Y 則表示為個別像素分群後的號碼(例：y = 1 表此像素為第 1 群)。

$$\text{Var}(X) = E[\text{Var}(X|Y)] + \text{Var}(E[X|Y]) \quad (4)$$

從式子(4)中可以看到，X 的變異量(Var(X))為 X 在 Y 條件下其變異量的期望值(E[Var(X|Y)])再加上 X 在 Y 條件下其期望值的變異量(Var(E[X|Y]))，其中 E[Var(X|Y)]在這裡也可代表為區塊像素在分群後，分別用各群的重建值(各群平均數)重建下的 Mean squared error (MSE)，而 Var(E[X|Y])則代表著每點像素之重建值間的變異量，因此從表 9 中的數據可以看到當 X 的變異量在有預測(Prediction)時，其變異量較無預測時(No prediction)小，也就是在經過預測後，區塊內的變異量確實下降了，然而在 E[Var(X|Y)](同時代表 MSE)的數值可以發現，無預測的 MSE 卻小於有預測的 MSE，其中這是因為 Var(E[X|Y])(每個重建值間的變異量)在有預測時會遠小於無預測的，也就是代表著無預測時的重建值間是很分散的，而在經過預測後其重建值也變得較為集中，因此從以上數據可得知，區塊經過預測後變異量(Var(X))確實下降了，然而各重建值間的變異量(Var(E[X|Y]))也同時並更大幅度下降，因此反而使得其 MSE (E[Var(X|Y)])上升，也就是從圖 25 的分布圖中可以看到，預測前的分布較集中於兩群，但此兩群距離較遠使得其變異量上升，然而經過預測後雖然分布較為集中(變異量下降)，但其重建值也較為接近，反而各群內的值確較為分散，因此導致 MSE 上升。由於以上為單一區塊的例子，因此我們也平均了許多區塊在不同的重建值(Base color)

個數下其誤差量的影響，如圖 26 所示，我們可以看到當只有 1 個重建值時，其預測後的誤差量會小於無預測的誤差量(此時剛好同為變異量)，然而當增加重建值數量後，情況便會開始相反，無預測的誤差量會開始小於預測後的誤差量。

在位元花費方面，從圖 27 可以看到不管在幾個重建值個數下，無預測所需的位元均小於預測的所需位元，那是由於經過預測後，其重建值的範圍變大，因此造成壓縮每個重建值時所需的額外位元增加，根據以上分析誤差量和所需位元對不同重建值個數下的影響，不使用預測工具對於 BCIM 是比較有好處的，而且不管是從表 3 的實驗結果或是剛才的分析均得到相同結論，因此我們決定在使用 BCIM 的時候不使用預測後的影像結果，而直接對原始影像進行量化及進行編碼。

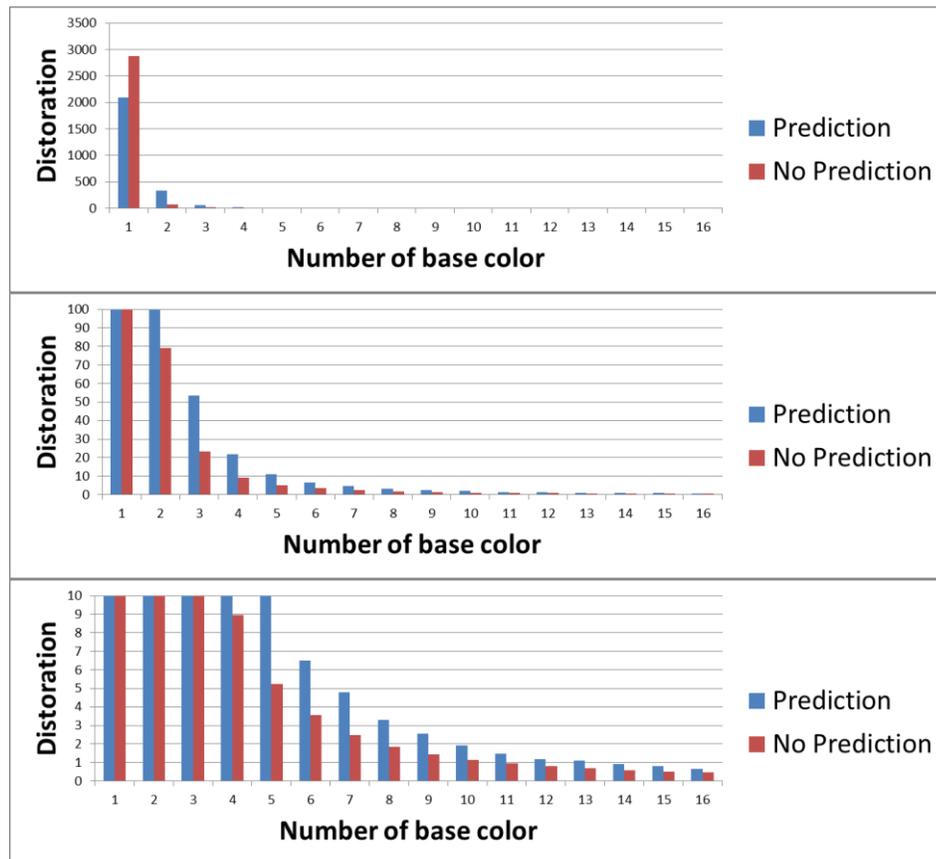


圖 26：預測和無預測在不同重建值個數下其誤差量變化

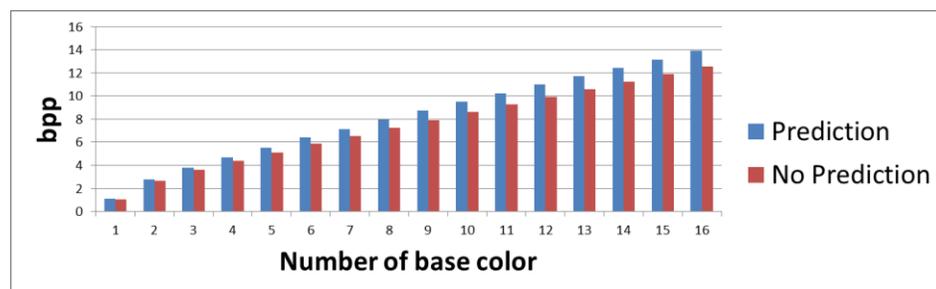


圖 27：預測和無預測在不同重建值個數下其位元花費

3.2 演算法概念與流程

從圖 2 的觀察可發現，區塊內可以將像素分成兩種類，一種為文字像素另一為圖像像素如圖 28 所示，針對文字像素將採用 BCIM 的方式編碼，而圖像像素因為包含許多不同顏色但存在某種關係而成為特殊平面，因此可使用方程式來表示其顏色變化。

針對圖像像素所組成特殊平面，根據其圖像像素的特性，本論文將使用兩種關係式表示，並決定其方程式。

◇ 確定性關係表示：漸層平面

這種方程式是針對其已經確定圖像像素間關係而設計的表示法，如漸層的線性平面，當確定為漸層的線性平面時，則可以使用簡單的線性方程式表達，以達到最佳的壓縮效能，而本文將採用漸層平面的方程式表示。

◇ 統計上關係表示：Shape-adaptive DCT (SADCT)

此種方程式是針對無法決定單一圖像像素間關係而設計的表示法，由於無法確定單一區塊內圖像像素的關係，所以將採用統計上的方法，找出統計上可能的最佳表示法，如傳統編碼中針對自然影像所設計的 DCT，而本文將採用 SADCT 表示。

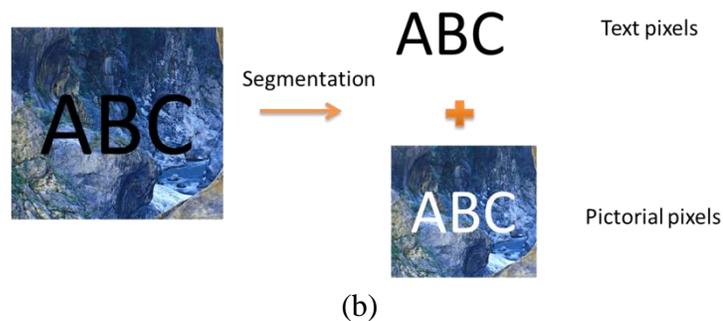
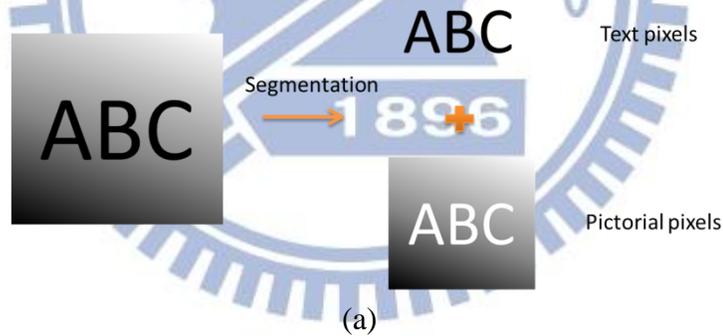


圖 28：視窗影像分成文字像素及圖像像素(a) 確定性關係表示圖像像素;(b) 統計上關係表示圖像像素;

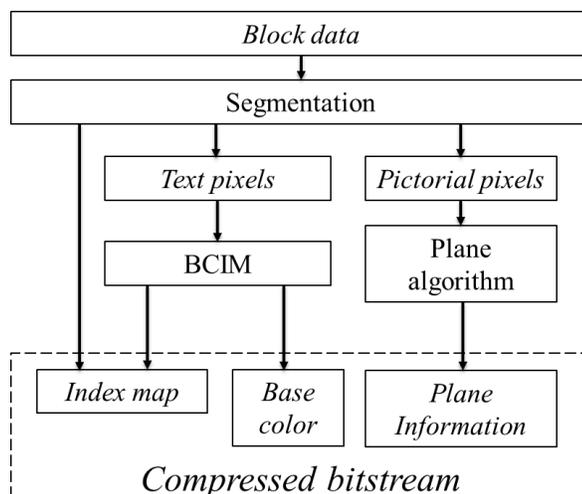


圖 29：演算法流程

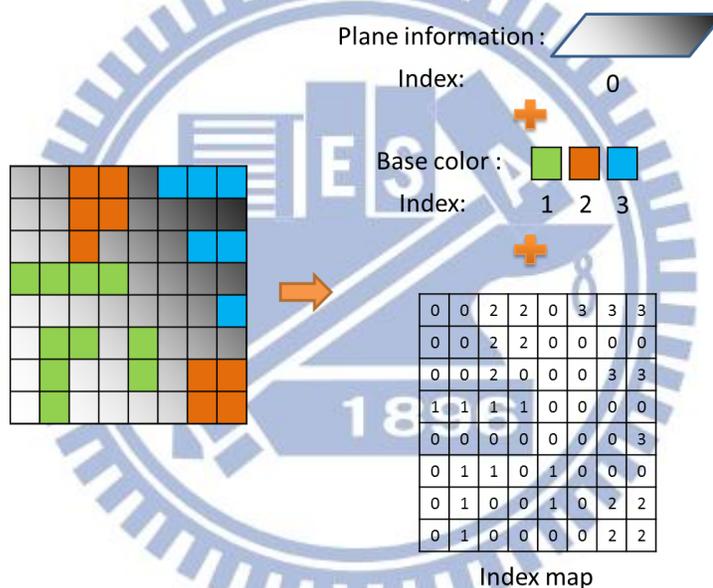


圖 30：加入特殊平面的編碼架構

演算法流程

上述介紹過了本方法的基本概念架構，因此將介紹實際的演算法流程，如圖29所示，首先會將影像區塊做切割，分成文字像素及圖像像素，針對圖像像素則做相對應之特殊平面演算法(確定性關係表示或統計上關係表示)，剩餘的文字像素則使用BCIM做量化求出分別的重建值，最後在分別編碼如圖30所示：特殊平面所需的相關資訊，索引圖及BCIM的重建值。

3.3 確定性關係表示特殊平面

在之前的概念中，我們將區塊內的圖像像素，使用方程式表示成一特殊平面，在本章

的演算法中，將採用確定性的方程式來表示，也就是根據此一平面的特性做此特殊平面表示，而此方程式將使用線性方程式來表示，如式子(5)所示， $f(x,y)$ 為該像素的重建值， x 、 y 分別表示該像素在區塊內的位置， a 、 b 、 c 表示此線性方程式的參數，因此傳送端只需編碼 a 、 b 、 c 這三個參數，解碼端即可重建出此一分層的資訊。

$$f(x,y) = ax + by + c \quad (5)$$

3.3.1 針對線性平面的分層切割演算法

在本章方法中，特殊分層的平面採用線性方程式表示，因此針對此一平面特性，設計了一套專門的切割演算法，由於公式(5)的方程式其變數的數量為三，因此當區塊內的像素數量大於三的時候，則可能無法完美表達所有像素，這時我們稱此方程式為under complete function，所以根據此一特性，我們可以使用特殊的切割方式找出符合線性方程式的平面及其 a 、 b 、 c 參數，這裡稱之為Linear Consensus (LC)的切割方式。

LC的主要概念為從其中一小子群的像素中求出其線性關係，之後再找出所有符合此線性關係的像素，如此就可求出滿足一線性關係的所有像素，然而每次子群所求出的像素均不一定相同，因此最後將決定含有最多像素的線性關係為所求之線性平面。

演算法流程

上述說明了此演算法的概念，以下將用步驟方式說明此演算法的完整流程

Step1：分出子群，且希望有一子群其均為漸層平面之像素

首先會先將所有像素收集，並計算一閾值(此處為平均數)為切割依據，之後將像素依閾值以上和閾值以下分成兩類，因為考量線性平面的部分有可能會介於兩非線性平面像素值之間，因此採用分四群的方式，也就是對第一次分完的兩群，再做一次閾值的切割，如圖31所示。

Step2：求出所有可能的線性平面

對所有子群做線性迴歸求出其線性關係，此時將求出四組線性方程式，再個別找出符合此線性方程式之像素，如圖32所示從第二群內的所有像素求出一線性方程式(虛線所示)，再依此方程式找出符合此方程式的所有像素(紅圈所標示的像素)，如此就可求出從一子群中找出一群符合線性方程式的像素。

Step3：決定最佳的線性平面

分別找出四組有線性關係的像素後，再比較此四組的像素數量，將選擇包含最多個數的像素群為此線性平面。

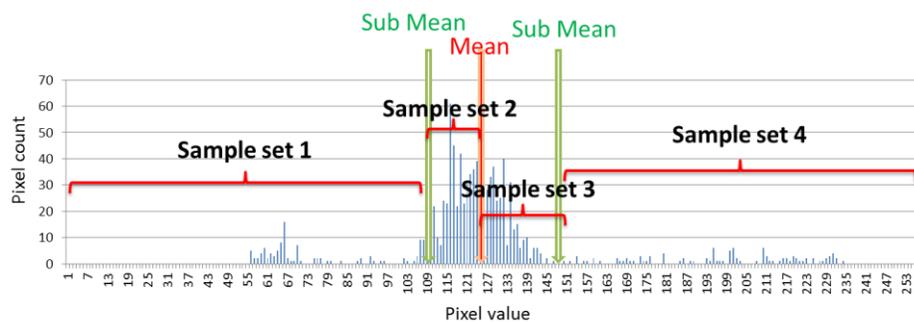


圖 31：區塊內像素值的直方圖

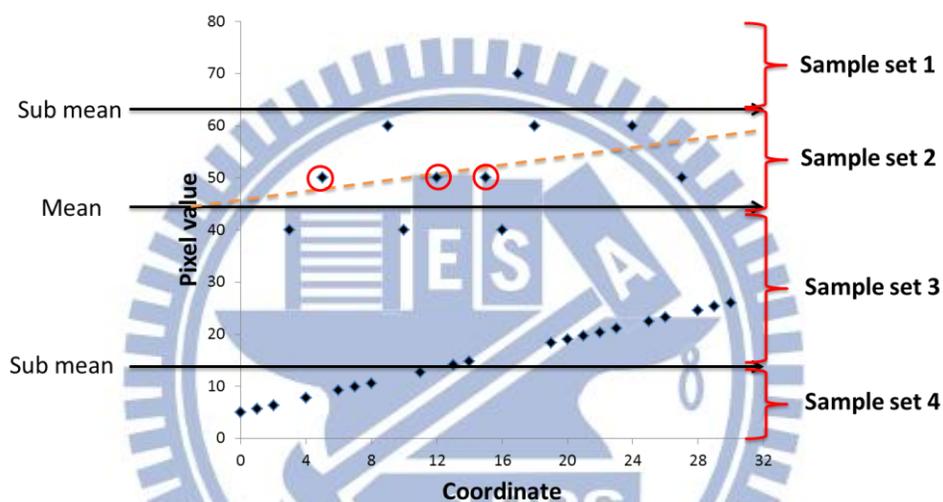


圖 32：Linear Consensus (LC)的切割方式

3.3.2 Entropy Coding

在演算法完成後，根據演算法的流程架構圖 29：演算法流程，除原本 BCIM 所需編碼的重建值，及共用的索引圖外，還需針對此線性平面編碼 a 、 b 、 c 之參數，在本文的方法中，均分別使用 1byte 來表示 a 、 b 、 c 之參數。

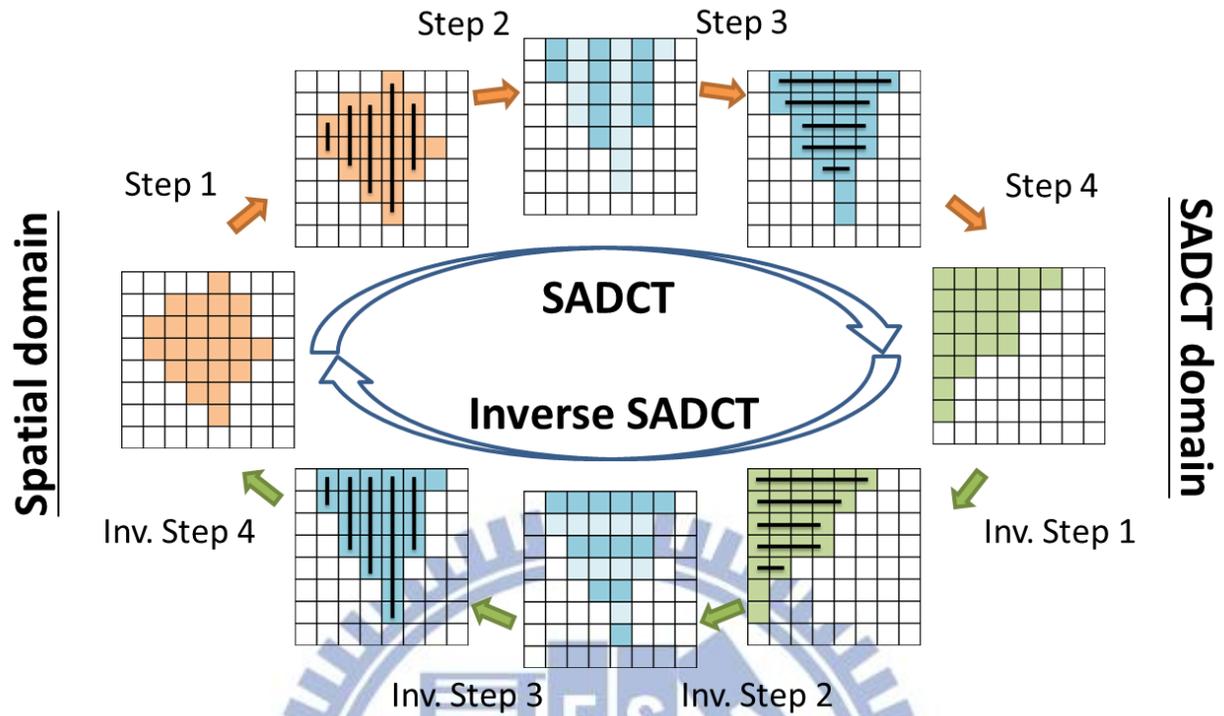


圖 33：SADCT 流程圖

3.4 統計上關係表示特殊平面

在前一節中，我們利用了確定性的方程式來表示特殊平面，但由於確定性的方程式所能表示的平面有限，而在本論文的方法中，只能表示漸層等較簡單的關係，因此為了能表示更複雜的特殊平面關係，此節將採用統計的方式表示此一平面，也就是認為此一平面上的像素會有一定上統計的關係，因此將使用 DCT 的方式表示此一平面，而由於非完整平面，在相關研究中有人提出使用 hole filling 或 Matching pursuit 等方式解決，但此類方法都會將其資料量膨脹，因此本論文將使用 MPEG4 中的 SADCT 演算法編碼此平面。

3.4.1 Shape-adaptive DCT (SADCT)

SADCT 是一種可以針對特殊形狀做 DCT 的方法，在傳統的 DCT 上，DCT 是必須使用在一個矩形的區塊上，然而 SADCT 則可以使用在非矩形的形狀上，並達到 DCT 的效果，因此本小節將介紹 SADCT 的演算法，並用來表示此一特殊平面。

圖 33 是 SADCT 基本的流程圖，左邊的區塊代表空間域(Spatial domain)上的影像數值，右邊的區塊表示經過 SADCT 轉換後頻率域(SADCT domain)上的結果，以下將說明正轉換及逆轉換的流程。

SADCT 正轉換(圖 33 上方由左至右)

Step1：進行垂直方向的 1D-DCT 轉換

由於其區塊內每行所計算的像素個數不一，因此會將區塊內每行分開計算，根據其不同的像素個數，分別做不同維度的 DCT，並分別求出其轉換係數(Coefficient)。

Step2：將不同維度的垂直 DCT 係數對齊

由於垂直 DCT 中每行的維度及像素位置都不相同，所以求出的 DCT 係數個數及係數位置也都不相同，然而為了能夠做水平方向的 DCT 轉換，必須安排哪些垂直係數一起做水平 DCT，因此這邊將所有垂直 DCT 係數往上推齊，使之集中在區塊最上方，也就是不同行間的 DC 和 DC、AC1 和 AC1、AC2 和 AC2...相互對齊。

Step3：進行水平方向的 1D-DCT 轉換

經由步驟 2 對齊之後，將對每列係數進行水平方向的 DCT 轉換，但由於每列的係數數量不相等，因此會根據每列的係數數量，做不同維度的水平 DCT。

Step4：SADCT 係數切齊區塊左邊

此時已經得到所有正轉換完的 SADCT 係數，但由於為了能貼近完整矩形 DCT 係數的分布(DCT 係數集中於區塊左上角)，因此將 SADCT 係數像左切齊至區塊左邊，此時方完成所有 SADCT 步驟。

SADCT 逆轉換(圖 33 下方由右至左)

逆轉換流程則是相對於正轉換做反向流程，即可還原得到原始空間域上的數值。

上述說明了 SADCT 的基本流程，然而在傳統的 SADCT 底下會有一些量化問題，因此 MPEG4 的 SADCT 為經過修改的，其核心的 DCT 不同於以往傳統的 SADCT，並且對 DC 值做了一些特別處理，以下將介紹其相關演算法。

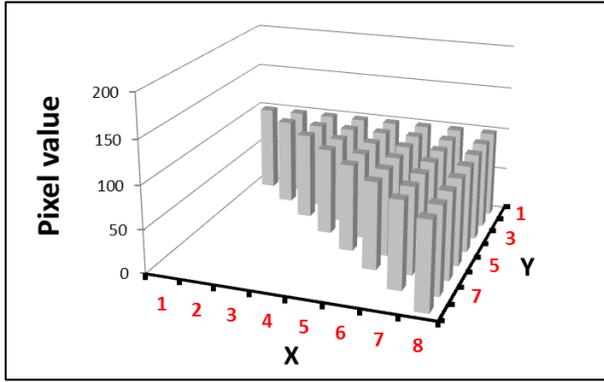
首先傳統 SADCT 的核心 DCT 為式子(6)，但由於 Scaling factor (如式子(6)中的 $2/N$ 及式子(7)中的 $\sqrt{2/N}$) 的設計得此 DCT 的基底不為 normalize，因此會根據不同維度的 DCT 造成係數有不同的倍數效果，且 SADCT 中每行間的像素個數不一，使得不同行間做完垂直 DCT 後會有不同的係數放大效果，第二次的水平 DCT 也會有同樣問題，因此使用式子(6)的核心 DCT，會使得在 SADCT 頻率域(SADCT domain)上的係數有不同的放大效果，所以當在給定相同的量化後，便會使得逆轉換回來後其空間域上的像素會有不一致的量化誤差，因此 MPEG4 的 SADCT 其核心 DCT 便改成式子(7)，此 DCT 為 normalize 的 DCT，也就表示不管核心 DCT 長度為多少，其基底長度均為 1，如此便可解決頻率域上得到相同量化誤差後造成空間域上誤差不同的情況。

$$\begin{aligned}
\text{Forward: } X_k &= \frac{2}{N} * \sum_{n=0}^{N-1} x_n DCT_N(n, k) \\
\text{Inverse: } x_n &= \sum_{k=0}^{N-1} X_k DCT_N(n, k)
\end{aligned} \tag{6}$$

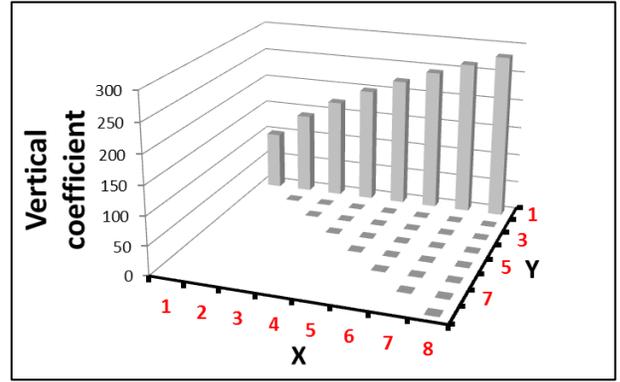
$$\begin{aligned}
\text{Forward: } X_k &= \sqrt{\frac{2}{N}} * \sum_{n=0}^{N-1} x_n DCT_N(n, k) \\
\text{Inverse: } x_n &= \sqrt{\frac{2}{N}} * \sum_{k=0}^{N-1} X_k DCT_N(n, k)
\end{aligned} \tag{7}$$

$$\begin{aligned}
DCT_N(n, k) &= c_o \cdot \cos\left(k\left(n + \frac{1}{2}\right)\frac{\pi}{N}\right) \\
\text{with } c_o &= \sqrt{1/2} \text{ if } k = 0 \text{ and } c_o = 1 \text{ elsewhere}
\end{aligned} \tag{8}$$

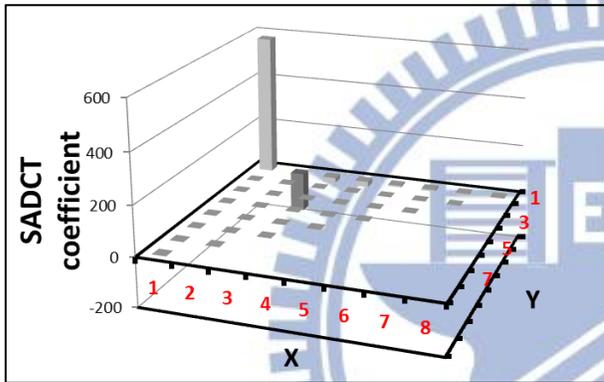
但將核心 DCT 改成 normalize 的 DCT 後，仍會存在一些問題，首先考慮當所有像素值均為相同時之例子如圖 34(a)所示，也就是在完整區塊時只存在 DC 值時的情況，但由於 SADCT 中每行的像素不一，所以當作完圖 34(b)的垂直 DCT 後，每一行的 DC 係數會因像素個數而不相同，之後作完水平 DCT 時如圖 34(c)，便會產生一些 AC 係數，而當再經過量化成圖 34(d)時，DC 及 AC 係數均會被量化，因為 AC 係數產生之真，所以在經由水平方向的逆轉換 DCT 轉成圖 34(e)時，便會使得垂直係數間的波形被改變，因此當最後重建成圖 34(f)時，每行間的值會不一定相同，也就是當原本只有一個相同像素值情況下，經由量化重建後卻會產生不同的像素值。為了預防以上情況，因此針對 DC 值的計算並不是直接經由圖 33 流程得到，而是採用分離 DC 的方式，如圖 35 所示先將所有像素值取平均數，並把所有像素扣除其平均值，再進行圖 33 的 SADCT 流程，最後將 DC 值用平均數取代，而逆轉換則是如圖 36 所示，先將 DC 值提出，並將 DC 設 0 做逆轉換，之後對轉換後的空間域，根據其重建完的值做一個 DC 的矯正演算法，此演算法的目的為填補因為使用平均數取代 DC 值所造成的差值，最後再將傳送的 DC 值(平均數)加上，得到最後還原值。使用分離 DC 的方法，可避免 DC 值因不同維度所造成的影響。



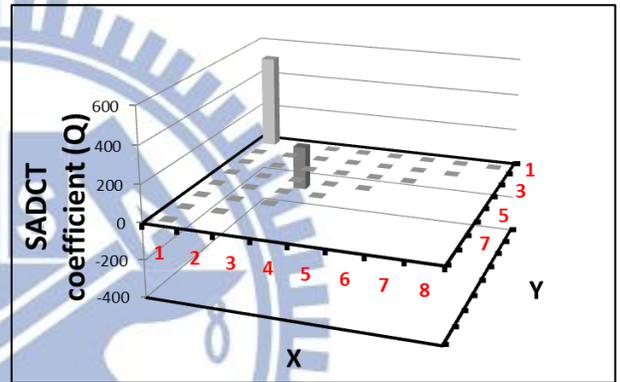
(a)



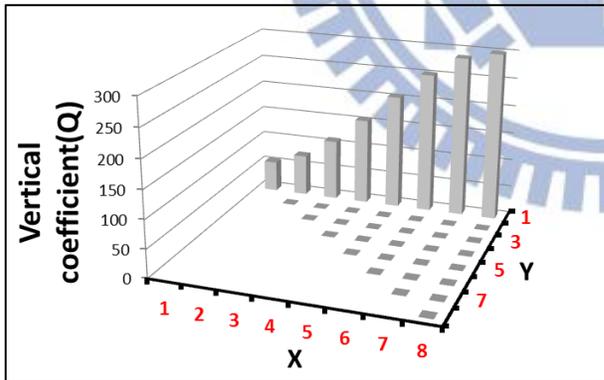
(b)



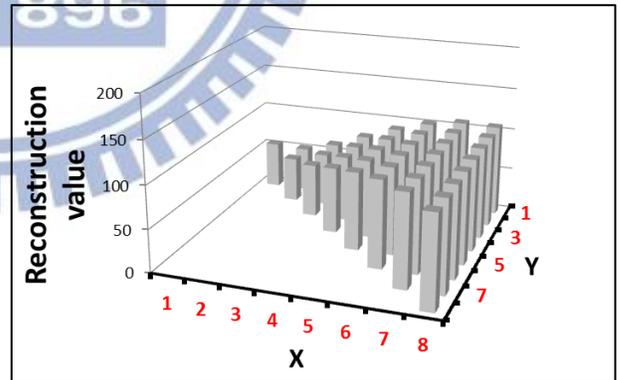
(c)



(d)



(e)



(f)

圖 34：Normalize SADCT 對 DC 的影響 (座標 X、Y 為像素之位置)

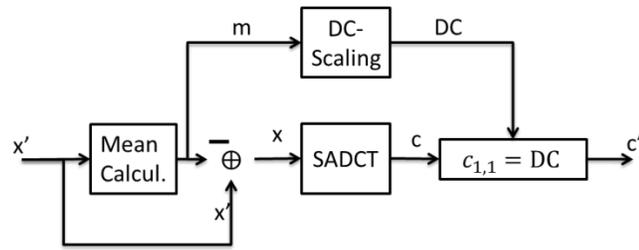


圖 35：分離 DC 流程圖(正轉換)

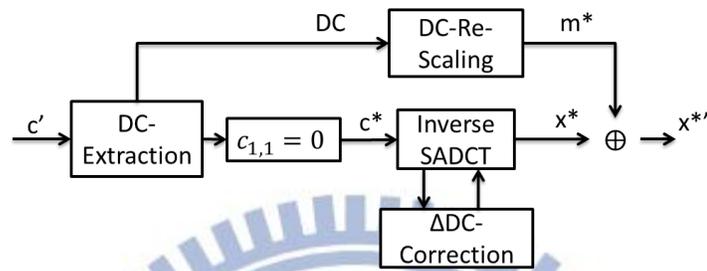


圖 36：分離 DC 流程圖(逆轉換)

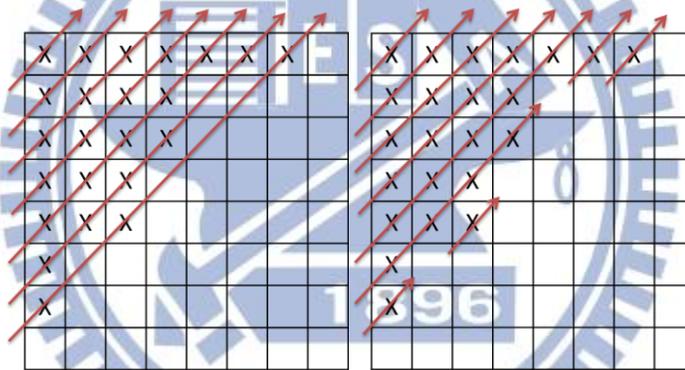


圖 37：左圖為原本掃描方式、右圖為修改過之掃描方式

3.4.2 Entropy Coding

在演算法完成後，根據演算法的流程架構圖 29，除原本 BCIM 所需編碼的重建值，及共用的索引圖外，還需針對特殊平面的資訊做編碼，在本節所代表的就是 SADCT 係數，而在本文方法中將採用原本矩形 DCT 係數的編碼方式做編碼，但由於 SADCT 所編碼的區域並不是整個區塊，因此在掃描 SADCT 係數的過程中會自動跳過原本就沒有係數的位置，以節省不必要的位元浪費。如圖 37 所示 X 為 SADCT 係數位置、箭頭為掃描方式，可以從左圖的原本矩形 DCT 掃描方式發現，會額外掃到原本就沒有 SADCT 係數的位置，因此根據其 SADCT 係數分布，如圖 37 右圖所示將會自動跳過其空白的係數位置。

第四章、實驗結果與討論

4.1 確定性關係(漸層平面)之實驗結果

本篇論文採用 HM7.0 作為實驗平台，並額外增加編碼工具於 HM7.0 中，比較的對象為 HM7.0 使用通用的測試設定[22]，其架構如表 10 所示。

Reference software	HM7.0
Quantization parameters	22, 27, 32, 37
Coding structure	Intra_HE10(All intra)
CU size	8x8~64x64
PU partition	Symmetric
TU partition	RQT
Intra prediction	DC, Planar, 33 Directions, DM, LM Chroma
Transform	DCT 4x4~32x32, Transform Skip 4x4
In-loop filter	De-blocking, SAO, ALF
Entropy coding	CABAC
Test sequence	Class F

表 10：實驗架構

4.1.1 切割演算法中不同閾值之選擇

在漸層平面的演算法中，本論文提出了一種切出漸層平面之演算法，其中有一部分為分割閾值之選擇，因此本實驗將比較以下兩種閾值選擇之方法，平均數(Mean)及中位數(Median)。

◇ Class F 影像測試

從測試影像為 Class F 的實驗數據表 11 及表 12 中可以發現，在閾值的選擇上使用平均數及中位數其客觀數據差不多，也就是採用中位數及平均數的方法幾乎都可以切出所需的像素，並搭配之後的演算法求出線性平面，由於平均數的 bits-rate saving 略多於中位數，且計算中位數時需要做排序演算法，因此本論文選擇採用平均數為閾值之選擇。

◇ 其他影像測試

本實驗為測試了在最近 HEVC 會議中，新討論的 10 個測試影像，如圖 38 所示，在新測試影像中的結果，從表 14 及表 15 可以看到，採用平均數為閾值的方法，其 bits-rate saving 仍略多於中位數，因此之後實驗均採用平均數之方法

4.1.2 BCIM 和漸層演算法之比較

◇ Class F 影像測試

本實驗將比較 BCIM 和漸層演算法，從表 12 和表 13 中測試影像為 Class F 的實驗數據，我們可以看到 BCIM 的位元率可節省 15.6% 左右，而漸層演算法也可節省約 15.2% 的位元率，因此在客觀數據上來看，BCIM 和漸層演算法可達到約差不多位元節省，且在 *SlideEditing* 這測試影像下，漸層演算法能比 BCIM 多節省約 1%，這也表示漸層演算法的確在有漸層背景的測試影像下能提供較佳的客觀壓縮效果，然而在其他測試影片，由於沒有漸層平面之特性，且表示一個單一顏色的線性平面所花費的位元會比 BCIM 的單一顏色還多，所以相較於 BCIM 所花的位元率會些微上升。

在主觀視覺方面，從圖 39 中可以清楚看到在 *SlideEditing* 的測試影像中，傳統 HM 的壓縮方法會導致文字有模糊的情況產生，而 BCIM 的方法雖然文字較為清晰，但仍會在背景部份產生明顯的量化失真，然而採用漸層平面的演算法除字體清晰外，因採用線性平面表示其背景關係，使得背景能清楚產生，不會如 BCIM 般產生量化失真。

◇ 其他影像測試

本實驗為測試了在最近 HEVC 會議中，新討論的 10 個測試影像，如圖 38 所示，在新測試影像中的結果，從表 15 及表 16 可以看到，整體來說 BCIM 約比漸層演算法多減省了約 1.4% 左右的位元，也就是在其他無漸層背景的測試影像下，用線性平面表示單一顏色相對於 BCIM 用單一重建值(Base color)，大約會差了 1% 左右的位元。

表 11：前景 BCIM、背景漸層平面、切割線性切割(中位數)之 bits-rate saving

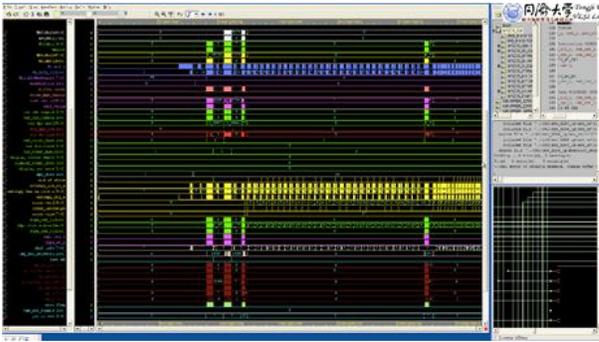
Class F	Y	U	V
<i>BasketballDrillText</i>	-2.2%	0.0%	-0.1%
<i>ChinaSpeed</i>	-15.9%	-12.6%	-9.4%
<i>SlideEditing</i>	-29.3%	-21.0%	-21.8%
<i>SlideShow</i>	-10.9%	-8.1%	-8.9%
All	-14.6%	-10.4%	-10.1%

表 12：前景 BCIM、背景漸層平面、切割線性切割(平均數)之 bits-rate saving

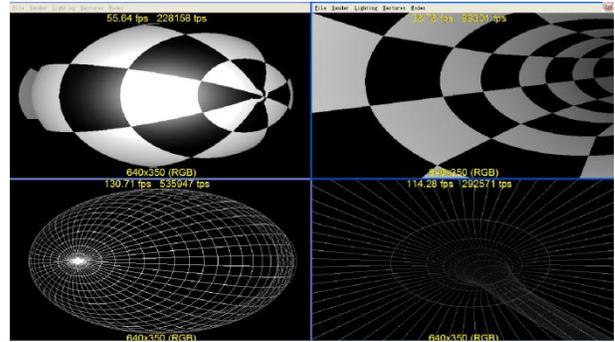
Class F	Y	U	V
<i>BasketballDrillText</i>	-2.2%	0.0%	-0.1%
<i>ChinaSpeed</i>	-16.0%	-12.6%	-9.4%
<i>SlideEditing</i>	-30.5%	-21.9%	-22.7%
<i>SlideShow</i>	-12.1%	-9.0%	-9.7%
All	-15.2%	-10.9%	-10.5%

表 13：BCIM 之 bits-rate saving

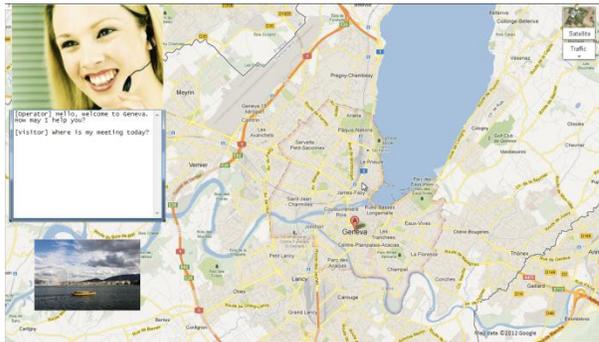
Class F	Y	U	V
<i>BasketballDrillText</i>	-2.6%	-0.7%	-0.8%
<i>ChinaSpeed</i>	-16.7%	-13.5%	-11.1%
<i>SlideEditing</i>	-29.4%	-21.2%	-22.3%
<i>SlideShow</i>	-13.5%	-10.4%	-10.5%
All	-15.6%	-11.4%	-11.2%



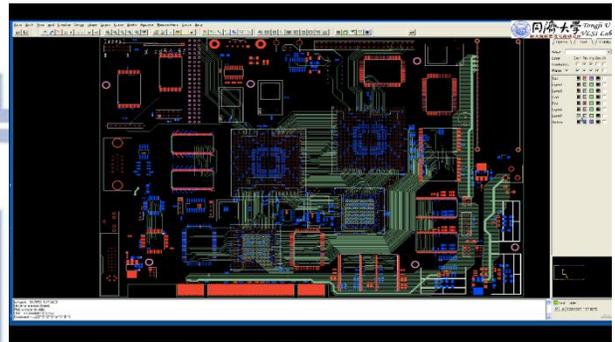
(a)



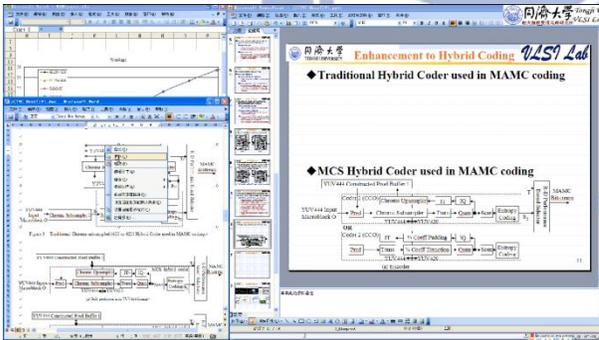
(b)



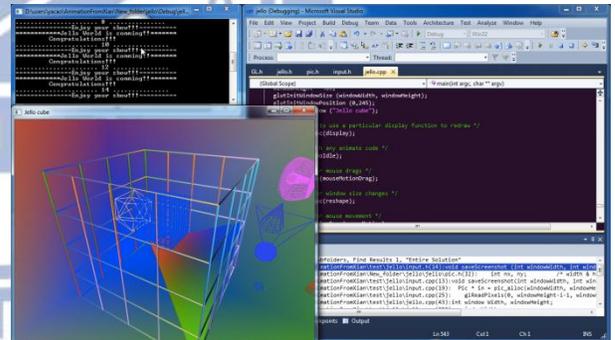
(c)



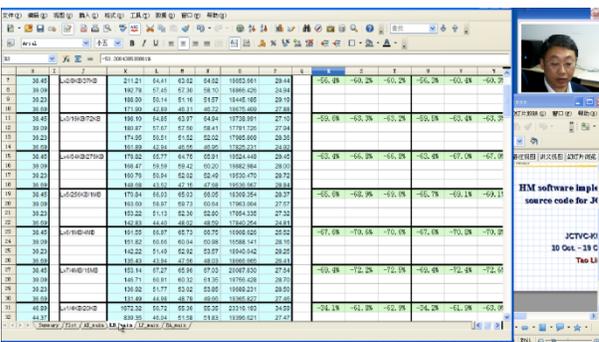
(d)



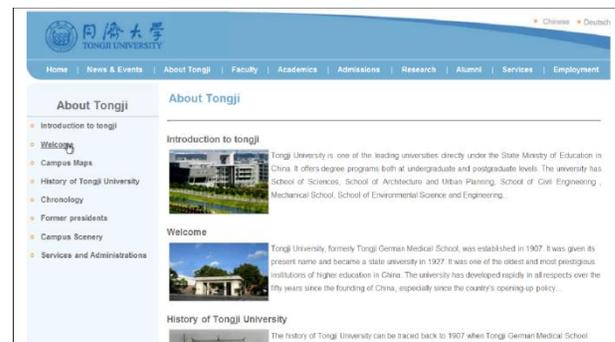
(e)



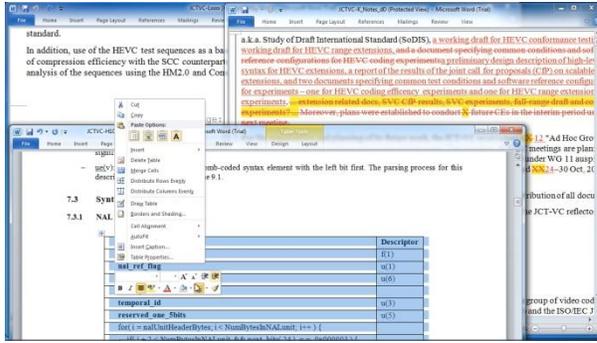
(f)



(g)



(h)



(i)



(j)

圖 38 : HEVC 中新討論的測試影像(a) *sc_cad_waveform*;(b) *sc_cg_twist_tunnel*;(c) *sc_map*;(d) *sc_pcb_layout*;(e) *sc_ppt_doc_xls*;(f) *sc_ppt_doc_xls*;(g) *sc_programming*;(h) *sc_video_conferencing_doc_sharing*;(i) *sc_web_browsing*;(j) *VenueVu*;

表 14 : 前景 BCIM、背景漸層平面、切割線性切割(中位數)之 bits-rate saving

Other sequence	Y	U	V
<i>sc_cad_waveform</i>	-31.7%	-18.6%	-18.1%
<i>sc_cg_twist_tunnel</i>	-54.9%	-46.0%	-46.4%
<i>sc_map</i>	-7.8%	-0.7%	2.1%
<i>sc_pcb_layout</i>	-25.1%	-13.9%	-12.3%
<i>sc_ppt_doc_xls</i>	-25.5%	-13.1%	-11.7%
<i>sc_programming</i>	-25.1%	-17.7%	-18.4%
<i>sc_video_conferencing_doc_sharing</i>	-30.5%	-20.1%	-17.9%
<i>sc_web_browsing</i>	-25.2%	-12.1%	-10.1%
<i>sc_wordEditing</i>	-27.1%	-20.0%	-17.9%
<i>VenueVu</i>	0.1%	0.0%	0.1%
All	-23.0%	-14.8%	-13.7%

表 15 : 前景 BCIM、背景漸層平面、切割線性切割(平均數)之 bits-rate saving

Other sequence	Y	U	V
<i>sc_cad_waveform</i>	-32.9%	-19.2%	-18.9%
<i>sc_cg_twist_tunnel</i>	-55.0%	-46.0%	-46.4%
<i>sc_map</i>	-8.9%	-0.3%	2.6%
<i>sc_pcb_layout</i>	-25.3%	-14.1%	-12.5%
<i>sc_ppt_doc_xls</i>	-27.2%	-13.4%	-11.7%
<i>sc_programming</i>	-25.4%	-17.9%	-18.6%
<i>sc_video_conferencing_doc_sharing</i>	-31.7%	-20.6%	-18.2%
<i>sc_web_browsing</i>	-32.4%	-15.2%	-12.3%
<i>sc_wordEditing</i>	-28.4%	-21.0%	-18.7%
<i>VenueVu</i>	0.1%	0.0%	0.1%
All	-24.3%	-15.2%	-14.1%

表 16：BCIM 之 bits-rate saving

Other sequence	Y	U	V
<i>sc_cad_waveform</i>	-34.1%	-20.9%	-20.5%
<i>sc_cg_twist_tunnel</i>	-54.2%	-46.5%	-46.7%
<i>sc_map</i>	-12.5%	-0.3%	1.7%
<i>sc_pcb_layout</i>	-27.4%	-16.2%	-14.8%
<i>sc_ppt_doc_xls</i>	-29.0%	-15.7%	-13.9%
<i>sc_programming</i>	-26.5%	-19.2%	-20.0%
<i>sc_video_conferencing_doc_sharing</i>	-32.8%	-21.2%	-19.0%
<i>sc_web_browsing</i>	-34.9%	-16.1%	-13.1%
<i>sc_wordEditing</i>	-30.7%	-22.1%	-19.6%
<i>VenueVu</i>	0.1%	0.1%	0.0%
All	-25.7%	-16.2%	-15.1%

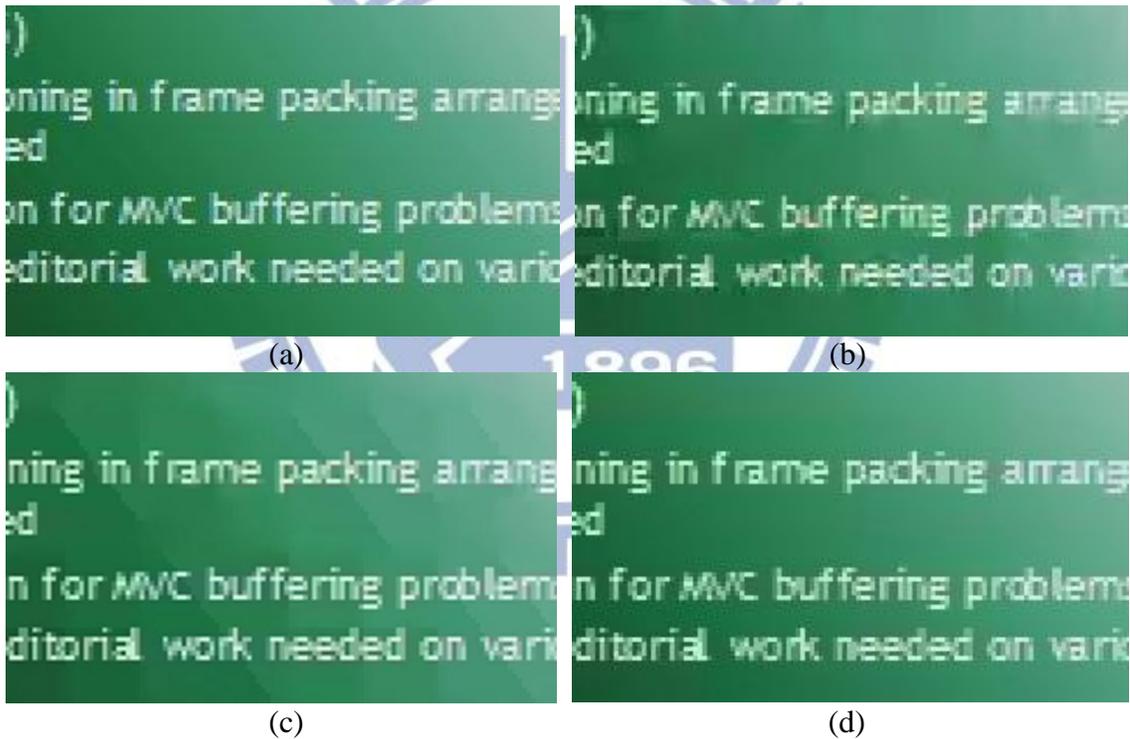


圖 39：SlideEditing 的主觀視覺(a) Original;(b) HM 7.0: 14308.00 kbps;(c) HM+BCIM: 14705.96 kbps;(d) HM+Linear plane: 12297.26 kbps;

and Information Systems Research Center, Room 704 & 720, 1001 Ta-Hsueh Rd., Hsinchu, 30010, Taiwan.

(a)

and Information Systems Research Center, Room 704 & 720, 1001 Ta-Hsueh Rd., Hsinchu, 30010, Taiwan.

(b)

and Information Systems Research Center, Room 704 & 720, 1001 Ta-Hsueh Rd., Hsinchu, 30010, Taiwan.

(c)

and Information Systems Research Center, Room 704 & 720, 1001 Ta-Hsueh Rd., Hsinchu, 30010, Taiwan.

(d)

and Information Systems Research Center, Room 704 & 720, 1001 Ta-Hsueh Rd., Hsinchu, 30010, Taiwan.

(e)

and Information Systems Research Center, Room 704 & 720, 1001 Ta-Hsueh Rd., Hsinchu, 30010, Taiwan.

(f)

圖 40 : *Gradient* 在 Qp37 之主觀視覺(a) Original;(b) HM7.0: 0.26 bits/pixel;(c) HM+BCIM: 0.27 bits/pixel;(d) HM+BCIM: 0.21 bits/pixel (e) HM+LinearPlane(LC): 0.24 bits/pixel;(f) HM+LinearPlane(GT): 0.24 bits/pixel;

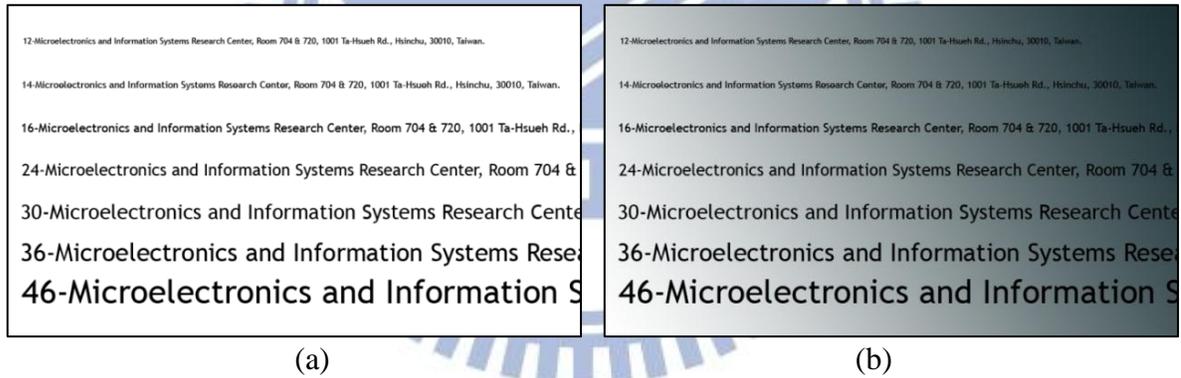


圖 41 : (a) *White*;(b) *Gradient*;

表 17 : 實驗方法說明

	切割演算法	文字像素	圖像像素	BCIM 重建值 (Base color)	平面參數 (a、b、c)
BCIM	無	BCIM		Pixel bits (10 bits)	無
LinearPlane (LC)	Linear Consensus	BCIM	漸層平面	Pixel bits (10 bits)	3x8 bits
LinearPlane (GT)	Ground truth	BCIM	漸層平面	Pixel bits (10 bits)	3x8 bits

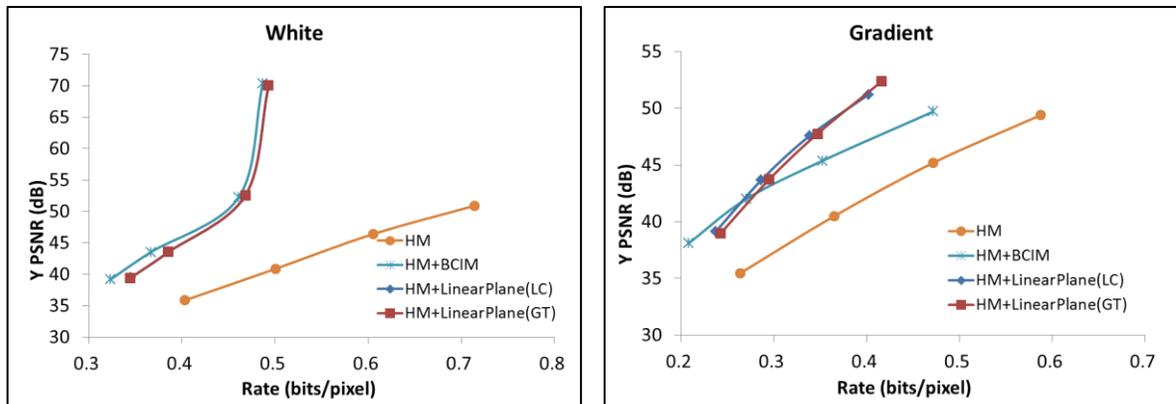


圖 42：漸層平面相關實驗的 Rate distortion curves

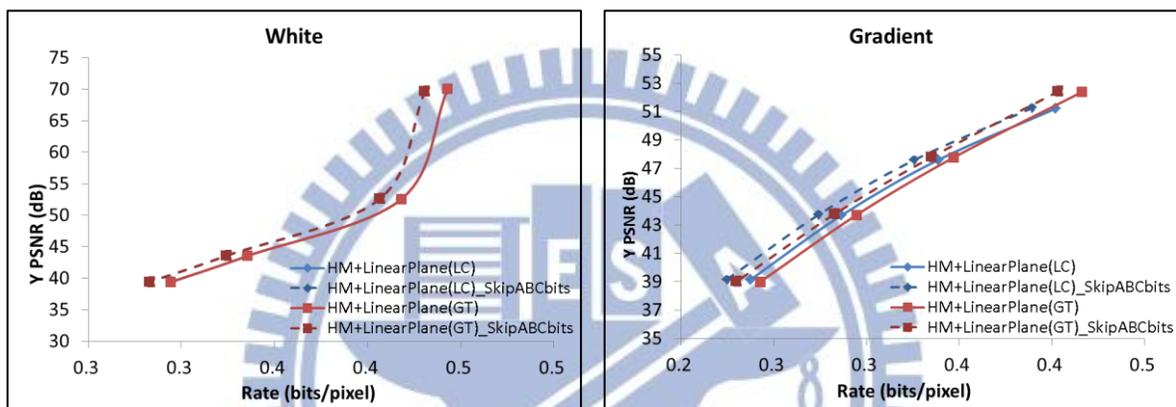


圖 43：漸層平面演算法中不傳送 a、b、c 參數的 Rate distortion curves

◇ 自製影像測試

本實驗測試了二張自製的圖片為圖 41(a) *White* 及圖 41(b) *Gradient*，*White* 是黑色文字搭配背景為全白的測試影像，*Gradient* 是黑色文字搭配背景為漸層的測試影像。圖 42 的 RD 中不同的曲線分別表示說明為，HM+X 表示為 HEVC 額外搭配 X mode，例 HM+ BCIM 表 HEVC 搭配 BCIM mode，本實驗中包含 HM、HM+BCIM、HM+LinearPlane (LC)、HM+LinearPlane (GT)，各方法說明如表 17。

從圖 42 中 *Gradient* 的 RD 曲線實驗結果可以發現，在有漸層背景的情況底下，HM+BCIM 演算法相較於 HM+LinearPlane 的演算法，在高品質需求時其壓縮效率會下降，這也就表示 BCIM 無法處理量化後顏色仍然過多的情況，因此這時使用漸層平面表示特定有相關性的顏色，則可減少索引圖位元的上升，而也可充分表現目前影像特性並減少失真。另外在主觀視覺上由於 BCIM 會將接近的值量化成單一數值，因此在低品質時，其背景會量化成單一顏色，就會有如圖 40(c)產生方塊般不連續現象，而採用漸層平面的演算法則不會有此現象。

另外從圖 42 中 *White* 的 RD 曲線實驗結果可以發現，在全白背景下，HM+BCIM 的客觀表現最佳，在低位元率時可提升約 10dB 左右，高位元率則可提升更多 dB，而採用

HM+LinearPlane 演算法，也約可同樣可提升 9dB，也就是在毫無特殊背景情況下，漸層平面和 BCIM 會差 1dB 左右，而這差距是由於在全白背景情況下，雖仍可使用漸層背景表示，但由於只有單一顏色，因此是不需要 a、b 參數，但在 HM+LinearPlane 的演算法下仍會傳送，所以造成所需位元上升，因而在效能上輸給 HM+BCIM，因此在此情況下可考慮搭配 BCIM 並動態選擇，或是在漸層平面下可動態選擇是否傳送 a、b 參數。

最後也觀察了當不傳送平面參數時，漸層平面所能多得到的效率，從圖 43 中 RD 曲線可以發現，其中 HM+LinearPlane(LC)_SkipABCbits 表示當漸層平面演算法時，不傳送 a、b、c 參數，從圖中可以發現如果不傳送時，則可比原本傳送時約增加 1 dB 左右的 PSNR，因此如果將 a、b、c 參數改進仍可有效提升編碼效率。

對於使用 Linear Consensus (LC)的切割方法，從圖 42 及圖 43 中的實驗數據中可發現，其切割效果幾乎和使用 Ground truth 差不多，因此採用 LC 分割演算法應是可以達到分割出平面背景的效果。

4.2 統計上關係(SADCT)之實驗結果

本篇論文採用 HM7.0 作為實驗平台，並額外增加編碼工具於 HM7.0 中，比較的對象為 HM7.0 使用通用的測試設定[22]，其架構如表 10 所示。

4.2.1 SADCT 表示前景文字像素

從圖 44 可以發現，將文字部分放大後，文字的邊緣會有一些模糊的部分，其顏色介於文字和背景之間，因此可以發現文字部分並不是單一顏色所組成，所以我們嘗試針對文字部分使用 SADCT，看是否能有效表示。

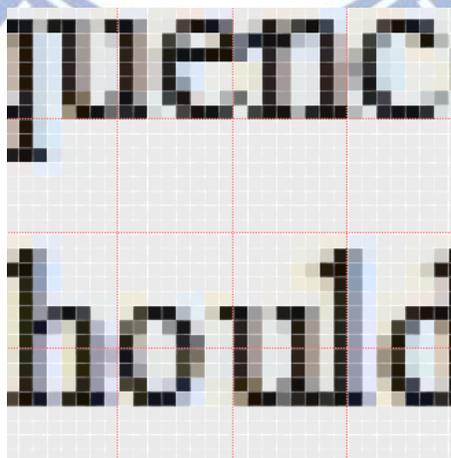


圖 44：SlideEditing 文字部分

表 18：前景 SADCT、背景漸層平面、切割線性切割(平均數)之 bits-rate saving

Class F	Y	U	V
<i>BasketballDrillText</i>	-1.0%	0.2%	0.1%
<i>ChinaSpeed</i>	-12.1%	-10.5%	-7.5%
<i>SlideEditing</i>	-15.9%	-12.4%	-13.8%
<i>SlideShow</i>	-3.6%	-3.5%	-3.7%
All	-8.2%	-6.6%	-6.2%

表 19：前景 SADCT、背景漸層平面、切割線性切割(平均數)之 bits-rate saving

Other sequence	Y	U	V
<i>sc_cad_waveform</i>	-19.2%	-13.6%	-13.1%
<i>sc_cg_twist_tunnel</i>	-43.3%	-38.1%	-38.2%
<i>sc_map</i>	-3.0%	0.5%	3.7%
<i>sc_pcb_layout</i>	-13.8%	-9.1%	-7.6%
<i>sc_ppt_doc_xls</i>	-12.0%	-6.7%	-5.5%
<i>sc_programming</i>	-15.9%	-13.5%	-13.4%
<i>sc_video_conferencing_doc_sharing</i>	-18.2%	-14.7%	-14.0%
<i>sc_web_browsing</i>	-13.7%	-5.7%	-3.3%
<i>sc_wordEditing</i>	-9.8%	-8.4%	-7.4%
<i>VenueVu</i>	0.1%	0.1%	0.1%
All	-13.5%	-9.9%	-9.0%

◇ Class F 影像測試

從表 18 中測試影像為 Class F 的實驗結果來看，此方法相較於 HM7.0 能節省約 8% 的位元率，也就是將前景與背景分開，並分開壓縮的確能帶來好處，然而從表 12 和表 18 中的比較可以發現，在背景使用漸層平面及相同切割方法的情況底下，前景使用 BCIM 相比於使用 SADCT 結果，約多節省了約 7% 左右的位元率，這結果顯示前景文字部分雖然含有一些模糊的像素，但是採用 BCIM 的方法仍是較有效率。

◇ 其他影像測試

本實驗為測試了在最近 HEVC 會議中，新討論的 10 個測試影像，如圖 38 所示，在新測試影像中的結果，從表 16 及表 19 可以看到，基本上結果同 Class F 的實驗結果，在前景的部分採用 BCIM 的方法仍是較有效率。

4.2.2 SADCT 表示背景圖像像素

從前面實驗發現，前景文字部分選用 BCIM 的壓縮效果較佳，因此本實驗將使用 SADCT 表示背景部分，且背景是較為複雜的圖像像素組成，而在 Class F 中的測試影像較少有這種圖像，且為了不因為切割演算法影響編碼結果，因此這邊實驗的影像將採用自製的影像如圖 45，並使用 Ground truth 為切割依據。

◇ Class F 影像測試

本實驗也嘗試使用漸層平面中所做的切割演算法(LC)做切割，並測試 Class F 的效能，從實驗結果表 15、表 20 及表 21 的比較可以看到，在 LC 所切割出來的背景不管使用 SADCT、Matching Pursuit 或漸層平面，其壓縮效能均差不多，那是因為採用 LC 所切割的平面為線性平面，且 DCT 也可表示出線性平面，因此所得到壓縮效能差不多，因此之後的實驗將採用自製影像如圖 45，並使用 Ground truth 為切割依據。

表 20：前景 BCIM、背景 SADCT、切割線性切割(平均數)之 bits-rate saving

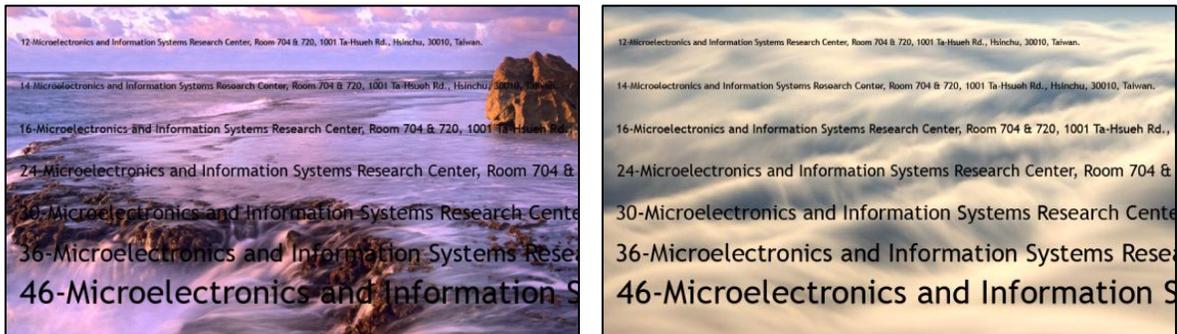
Class F	Y	U	V
<i>BasketballDrillText</i>	-1.9%	0.5%	0.4%
<i>ChinaSpeed</i>	-17.1%	-12.7%	-9.6%
<i>SlideEditing</i>	-29.7%	-18.7%	-19.5%
<i>SlideShow</i>	-11.7%	-8.4%	-8.4%
All	-15.1%	-9.8%	-9.3%

表 21：前景 BCIM、背景 Matching Pursuit、切割線性切割(平均數)之 bits-rate saving

Class F	Y	U	V
<i>BasketballDrillText</i>	-1.9%	0.5%	0.4%
<i>ChinaSpeed</i>	-17.4%	-12.7%	-9.4%
<i>SlideEditing</i>	-30.5%	-18.5%	-19.3%
<i>SlideShow</i>	-11.7%	-8.5%	-8.4%
All	-15.4%	-9.8%	-9.2%

◇ 自然背景圖樣的影像測試

本實驗測試了二張自製的圖片為圖 45(a) *Natural_ocean* 及圖 45(b) *Natural_cloud*，這兩張影像其背景均為自然圖像，而前景為文字混合影像，其中 *Natural_ocean* 的自然背景主觀上看來較 *Natural_cloud* 複雜。圖 47 的 RD 中不同的曲線分別表示說明為，HM+X 表示為 HEVC 額外搭配 X mode，例 HM+BCIM 表 HEVC 搭配 BCIM mode，本實驗中包含 HM、HM+BCIM、HM+LinearPlane (LC)、HM+LinearPlane (GT)、HM+SADCT (GT)、HM+Matching Pursuit (GT)，各方法說明如表 17 及表 22。



(a)

(b)

圖 45：(a) *Natural_Ocean*; (b) *Natural_cloud*;

表 22：實驗方法說明

	切割演算法	文字像素	圖像像素	BCIM 重建值 (Base color)	平面係數 (Coefficient)
BCIM	無	BCIM		Pixel bits (10 bits)	無
SADCT (GT)	Ground truth	BCIM	SADCT	Pixel bits (10 bits)	HM
Matching Pursuit (GT)	Ground truth	BCIM	Matching Pursuit	Pixel bits (10 bits)	HM

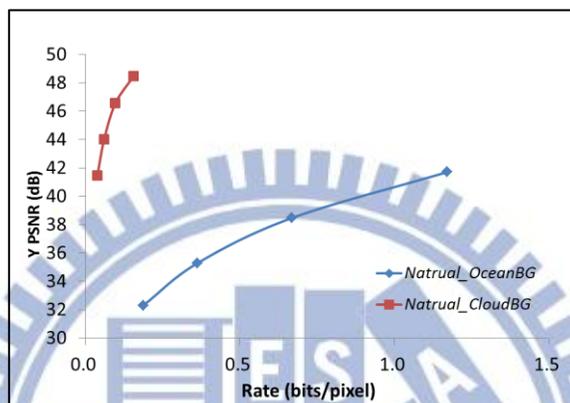


圖 46：Natural_Ocean 及 Natural_cloud 之背景使用 HM 壓縮的 Rate distortion curves

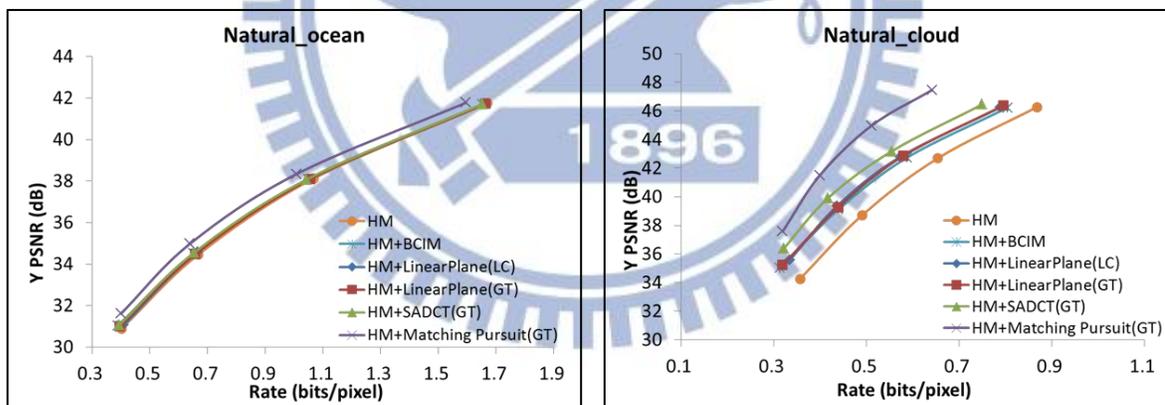


圖 47：自然背景影像實驗的 Rate distortion curves

首先從圖 46 的 RD 曲線中可以發現，在同樣使用 HM 做壓縮時，*Natural_cloudBG* (*Natural_cloud* 之背景圖)的 RD 曲線比 *Natural_oceanBG* (*Natural_ocean* 之背景圖)還要陡，也就表示 *Natural_ocean* 的背景圖樣比 *Natural_cloud* 還要複雜。從圖 47 中 *Natural_ocean* 的 RD 曲線可看到，在背景的自然影像較為複雜的情況下，HM+BCIM 及 HM+SADCT (GT) 均無法有效提升 PSNR，而採用 HM+Matching Pursuit (GT) 的方式，可提升約 0.5dB 的 PSNR，然而對於自然背景較簡單的 *Natural_cloud*，從其圖 47 中 *Natural_cloud* 的 RD 曲線可看到，HM+Matching Pursuit (GT) 即可大幅上升約 5dB 的 PSNR，HM+SADCT (GT) 也可上升了 3dB 的 PSNR，HM+BCIM 則約 2dB 左右，因此針對自然背景採用 DCT 式的編碼方式，仍能帶來一定的好處，且在較為簡單的自然背景下，其得到的好處更多，最後雖然 HM+SADCT (GT)

及 HM+Matching Pursuit (GT)均能帶來壓縮好處，但從實驗數據看來 HM+Matching Pursuit (GT)的編碼效能是比 HM+SADCT 的編碼方式有效。

另外從主觀視覺上來看，從圖 48 中可看到在低位元率的表現上，HM+Matching Pursuit (GT)的表現較為自然，而 HM+BCIM 因為其直接量化像素亮度值方式，所以導致背景圖出現塊狀的失真結構，另外 HM+SADCT (GT)在主觀視覺上也產生了些橫向不連續的失真效果，因此之後我們將討論 SADCT 的問題。

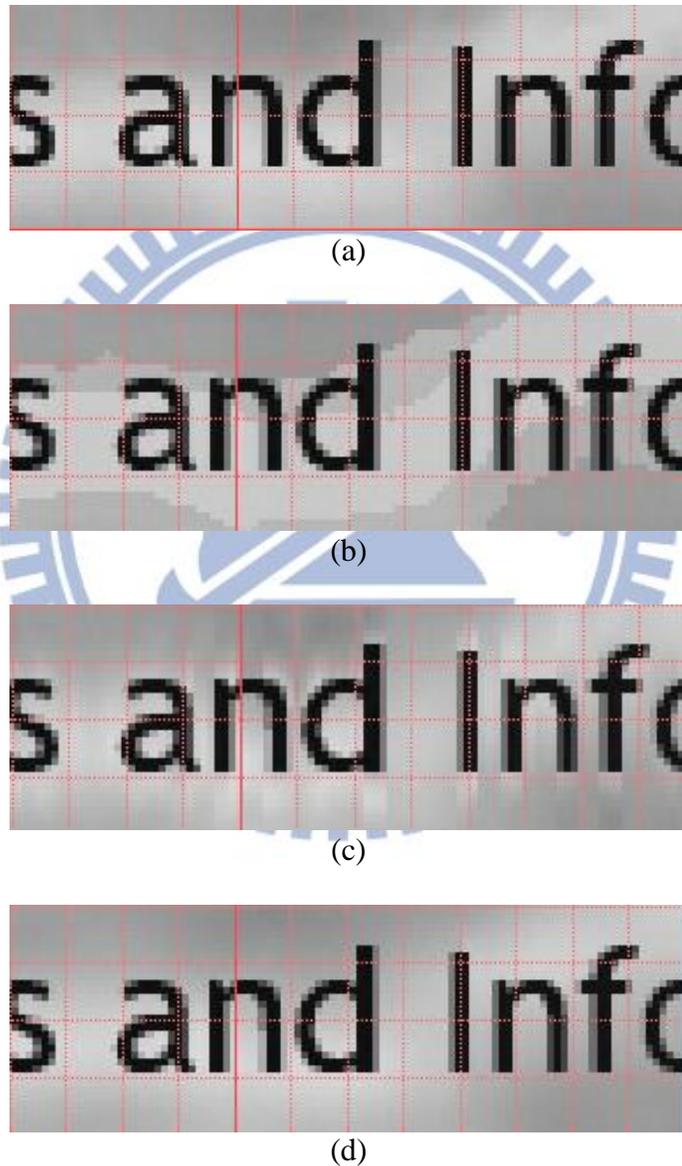


圖 48: *Natural_cloud* 在 Qp37 之主觀視覺(a) Original;(b) HM+BCIM 7.0: 0.31 bits/pixel;(c) HM+SADCT: 0.32 bits/pixel;(d) HM+Matching Pursuit: 0.32 bits/pixel;



(a) (b)
圖 49 : (a) *Natural_cloud(black)*; (b) *Natural_cloud(white)*;

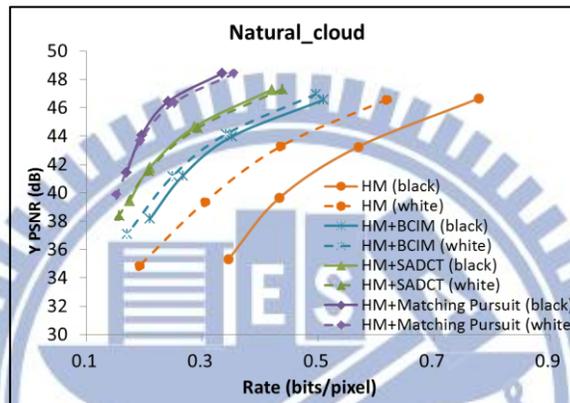


圖 50 : 不同文字亮度實驗的 Rate distortion curves

◇ 不同文字亮度的影像測試

本實驗將比較相同自然圖像背景下，不同的文字亮度對於 SADCT、Matching Pursuit 及 BCIM 的壓縮效果影響，測試的圖片為圖 49(a) *Natural_cloud(black)* 及圖 49(b) *Natural_cloud(white)*，這二張影像均為相同的自然背景，但分別搭配不同顏色的文字所組成，另外由於文字邊緣可能會影響實驗結果，因此以上圖片的文字均為單一顏色而沒有文字邊緣模糊的部分，圖 50 的 RD 中不同的曲線分別表示說明為，HM+X (Y) 表示為 HEVC 額外搭配 X mode 及在 Y 顏色文字的測試影像下，例 HM+BCIM (black) 表 HEVC 搭配 BCIM mode 及在黑色文字 (*Natural_cloud(black)*) 的測試影像下，本實驗中包含 HM (black)、HM (white)、HM+BCIM (black)、HM+BCIM (white)、HM+SADCT (black)、HM+SADCT (white)、HM+Matching Pursuit (black)、HM+Matching Pursuit (white)，各方法說明同之前的表 22。

從圖 50 中當文字為白色時 (*Natural_cloud(white)*)，其 RD 曲線相較於文字為黑色時 (*Natural_cloud(black)*) 的 RD 曲線，採用原始 HM 的編碼方式其 PSNR 便明顯提升不少，並減少了 HM+SADCT、HM+Matching Pursuit 及 HM+BCIM 相對於 HM 所提升的 PSNR，另外根據表 23、表 24 及表 25 可以發現當文字顏色為白色時，SADCT 及 Matching Pursuit 所占的編碼面積減少了，也就表示採用傳統 HM 編碼方式的面積增加了，因此這也代表著當文字和自然背景圖層的颜色越趨近時，將區塊切割成背景及前景並分開編碼，相對於不分割而整塊編碼所得到的客觀效率也會開始遞減，反之也就是當區塊內對比越高時，越需要

有分割前景及背景的編碼方式，另外從圖 51、圖 52 及圖 53 中各方法使用的區域中可發現，其中部分減少的面積，為原本使用分割方法的黑色文字區域，在文字改為白色時由於和背景顏色相似，因此選擇採用 HM 的編碼方式。

表 23：Matching Pursuit 於 *Natural_cloud(black)* 及 *Natural_cloud(white)* 所占的面積

	Qp22	Qp27	Qp32	Qp37
<i>Natural_cloud(black)</i>	49.68%	51.97%	53.69%	53.21%
<i>Natural_cloud(white)</i>	48.33%	49.82%	48.78%	43.97%

表 24：SADCT 於 *Natural_cloud(black)* 及 *Natural_cloud(white)* 所占的面積

	Qp22	Qp27	Qp32	Qp37
<i>Natural_cloud(black)</i>	41.37%	47.11%	49.57%	51.89%
<i>Natural_cloud(white)</i>	37.28%	44.01%	44.90%	41.86%

表 25：BCIM 於 *Natural_cloud(black)* 及 *Natural_cloud(white)* 所占的面積

	Qp22	Qp27	Qp32	Qp37
<i>Natural_cloud(black)</i>	18.28%	24.60%	29.69%	36.13%
<i>Natural_cloud(white)</i>	18.40%	21.78%	23.89%	24.16%



圖 51：Qp37 時 Matching Pursuit 於 *Natural_cloud(black)* 及 *Natural_cloud(white)* 所使用的區域(綠色區域)

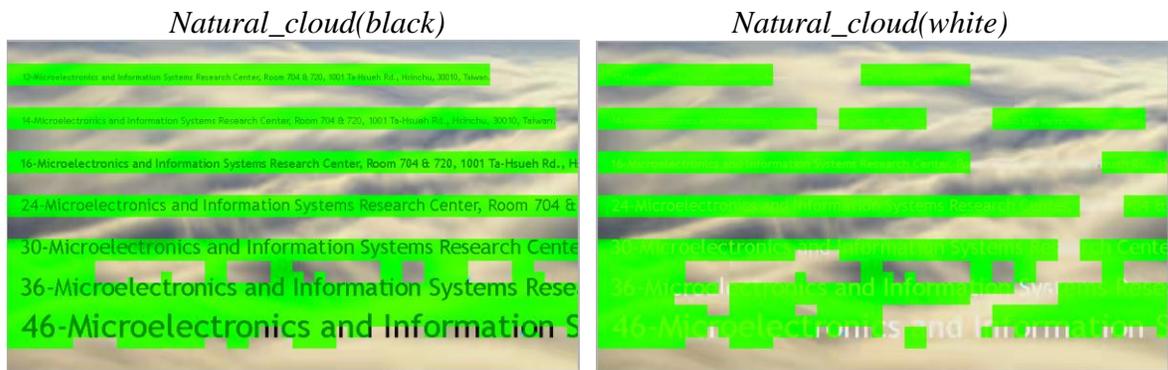


圖 52：Qp37 時 SADCT 於 *Natural_cloud(black)* 及 *Natural_cloud(white)* 所使用的區域(綠色區域)

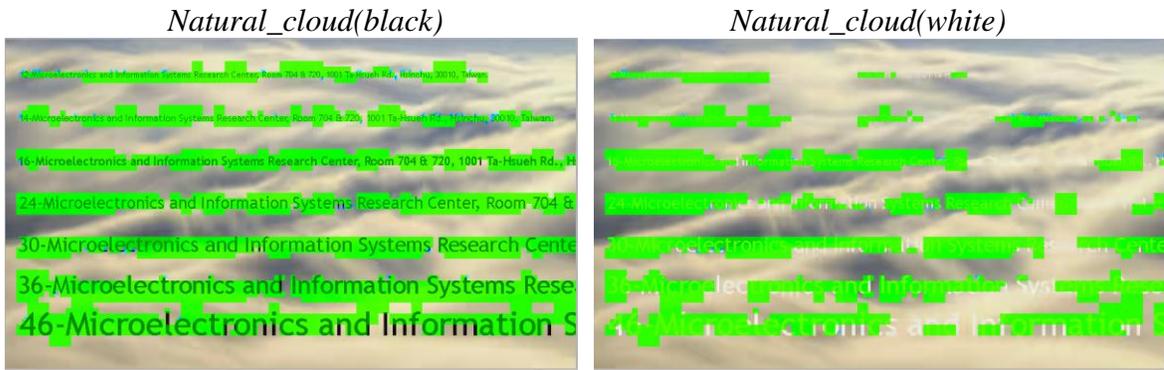
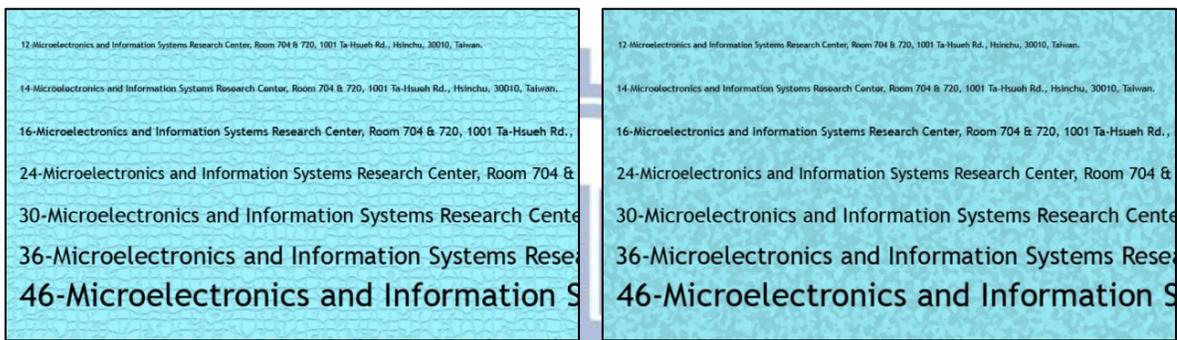


圖 53：Qp37 時 BCIM 於 *Natural_cloud(black)* 及 *Natural_cloud(white)* 所使用的區域(綠色區域)



(a) (b)

圖 54：(a) *Tile*; (b) *Water waves*;

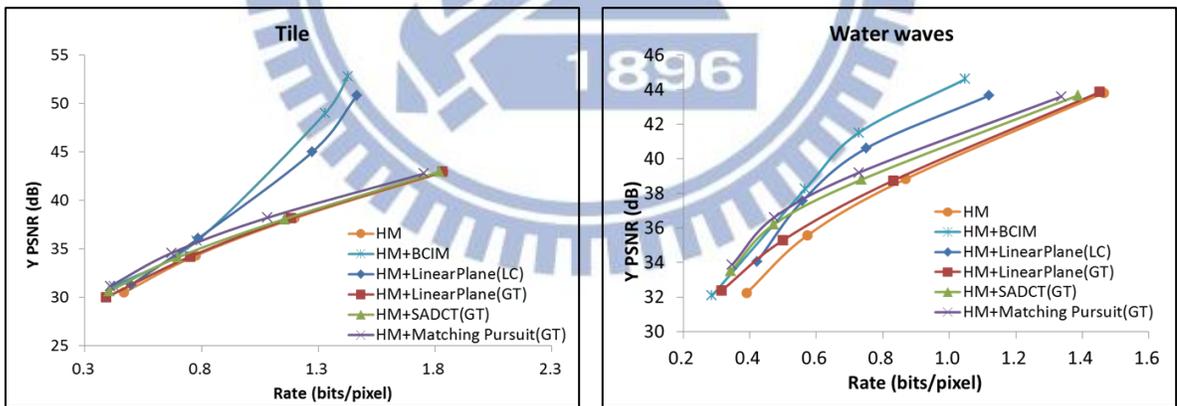


圖 55：背景紋理實驗的 Rate distortion curves

◇ 特殊背景紋理的影像測試

本實驗將比較在電腦產生的特殊紋理背景下如圖 54(a) *Tile* 及圖 54(b) *Water waves*，SADCT、Matching Pursuit 及 BCIM 的壓縮效果，圖 55 的 RD 中不同的曲線分別表示說明為，HM+X 表示為 HEVC 額外搭配 X mode，例 HM+BCIM 表 HEVC 搭配 BCIM mode，本實驗中包含 HM、HM+BCIM、HM+LinearPlane (LC)、HM+LinearPlane (GT)、HM+SADCT (GT)、HM+Matching Pursuit (GT)，各方法說明如表 17 及表 22。

從圖 55 中 *Tile* 及 *Water waves* 的 RD 曲線可清楚看到，HM+BCIM 在高品質需求時其表現相對較好，那是由於電腦生成的背景，其不同顏色較少，且富含特殊的紋路樣式，然而在低品質需求時，和其他演算法的表現就差不多，且從圖 56 的主觀視覺中，HM+BCIM 由於其量化特性關係，會使背景產生特殊的失真紋路，反而使用 HM+Matching pursuit 或 HM+SADCT 這類 DCT 轉換後所造成的失真，在主觀視覺上看起來比較自然。

另外從圖 55 中 *Tile* 及 *Water waves* 的 RD 曲線中可以發現，HM+LinearPlane (GT) 並無法有效提供壓縮效能，那是由於電腦產生的特殊紋理背景並無法使用一漸層平面表示，但採用相同編碼演算法 HM+LinearPlane (LC) 卻接近於 HM+BCIM，那是由於 HM+LinearPlane (LC) 是採用 Linear Consensus (LC) 的分割演算法，而此分割演算法在 *Tile* 及 *Water waves* 的影像下，其分割出的背景並非完整背景，而是如圖 58 所示，會將原本背景部分的某些區域歸類為前景，而留下幾乎為單一顏色的背景(綠色區域)，因此漸層平面就仍可表示分割完的背景像素，其餘原本的背景則和前景文字(桃紅色區域)一起採用 BCIM 壓縮。



圖 56： *Tile* 在 Qp37 之主觀視覺(a) Original;(b) HM+BCIM: 0.42 bits/pixel;(c) HM+LinearPlane (LC): 0.50 bits/pixel;(d) HM+LinearPlane (GT): 0.39 bits/pixel;(e) HM+SADCT (GT): 0.41 bits/pixel;(f) HM+Matching Pursuit (GT): 0.42 bits/pixel;

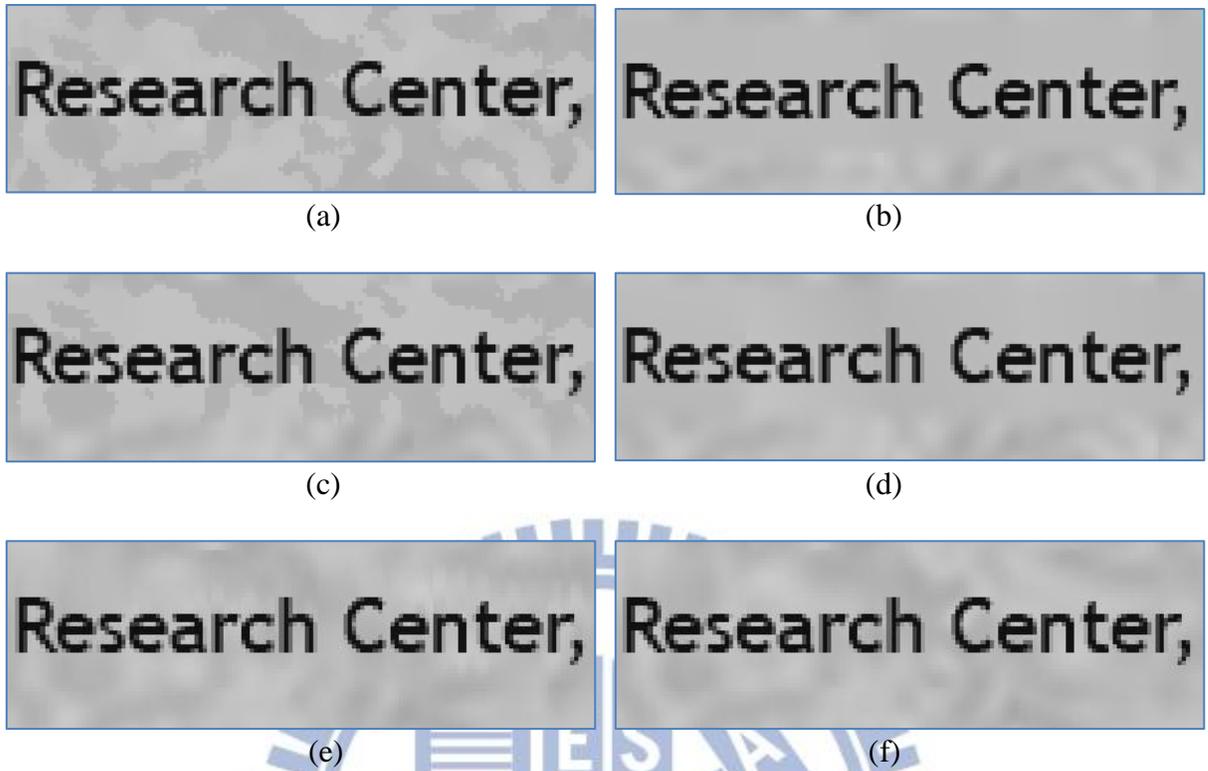


圖 57 : *Water waves* 在 Qp37 之主觀視覺(a) Original;(b) HM+BCIM: 0.29 bits/pixel;(c) HM+LinearPlane (LC): 0.42 bits/pixel;(d) HM+LinearPlane (GT): 0.32 bits/pixel;(e) HM+SADCT (GT): 0.34 bits/pixel;(f) HM+Matching Pursuit (GT): 0.35 bits/pixel;

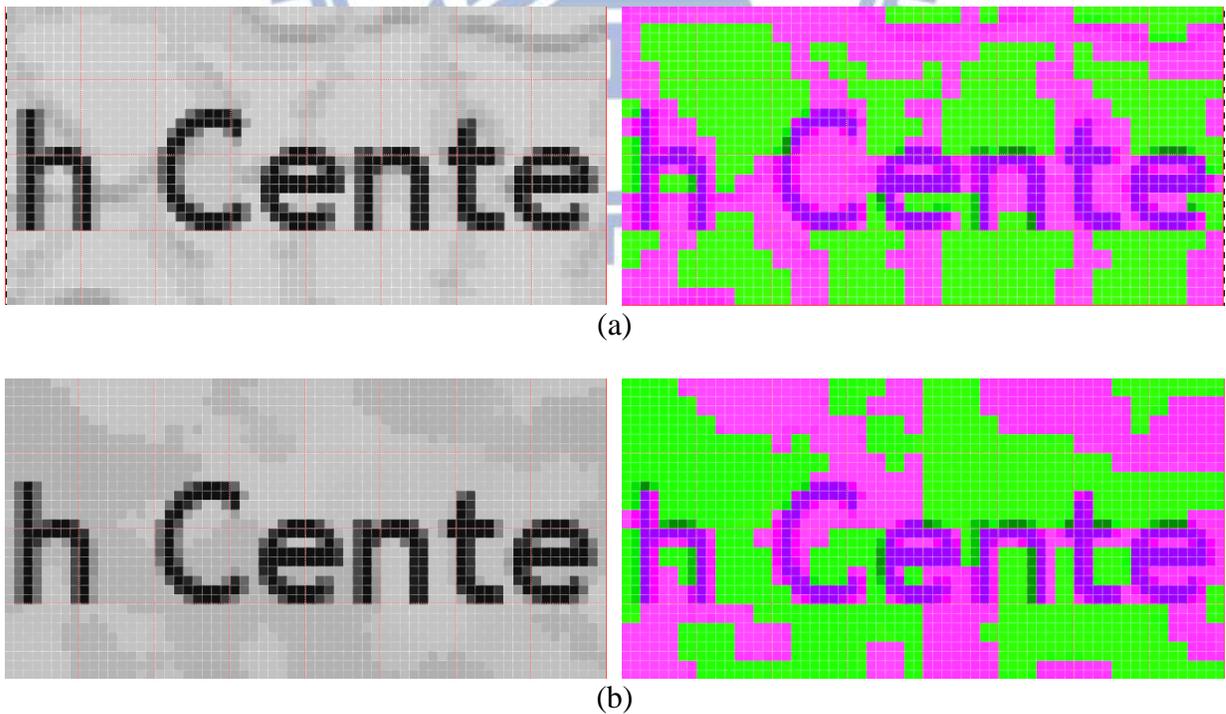


圖 58 : Linear Consensus (LC)分割出的桃紅色前景與綠色背景(a) *Tile*;(b) *Water waves*;

◇ 添加雜訊於紋理背景的影響測試

本實驗將比較在電腦產生的特殊紋理背景下如圖 54(a) *Tile* 及圖 54(b) *Water waves*，在添加不等程度的雜訊後其壓縮效果，圖 59 及圖 60 的 RD 中不同的曲線分別表示說明為，HM+X 表示為 HEVC 額外搭配 X mode，例 HM+BCIM 表 HEVC 搭配 BCIM mode，本實驗中包含 HM、HM+BCIM、HM+LinearPlane (LC)、HM+LinearPlane (GT)、HM+SADCT (GT)、HM+Matching Pursuit (GT)，各方法說明如表 17 及表 22。

從圖 59 及圖 60 的 RD 曲線中可以看到，當雜訊強度增加時使用視窗影像編碼的效率會開始下降，最後會幾乎和 HM 的效能差不多，也就是說雜訊會影響目前以上的視窗影像編碼效能，另外從圖 59(a) 中可發現當 SNR 為 48 dB 時，HM+BCIM 還能提供不錯的效能，然而 HM+LinearPlane (LC) 其效率已大幅下降，這表示當有雜訊時，本文所提供的 Linear Consensus (LC) 切割演算法可能會失效，導致其曲線遠離 HM+BCIM。

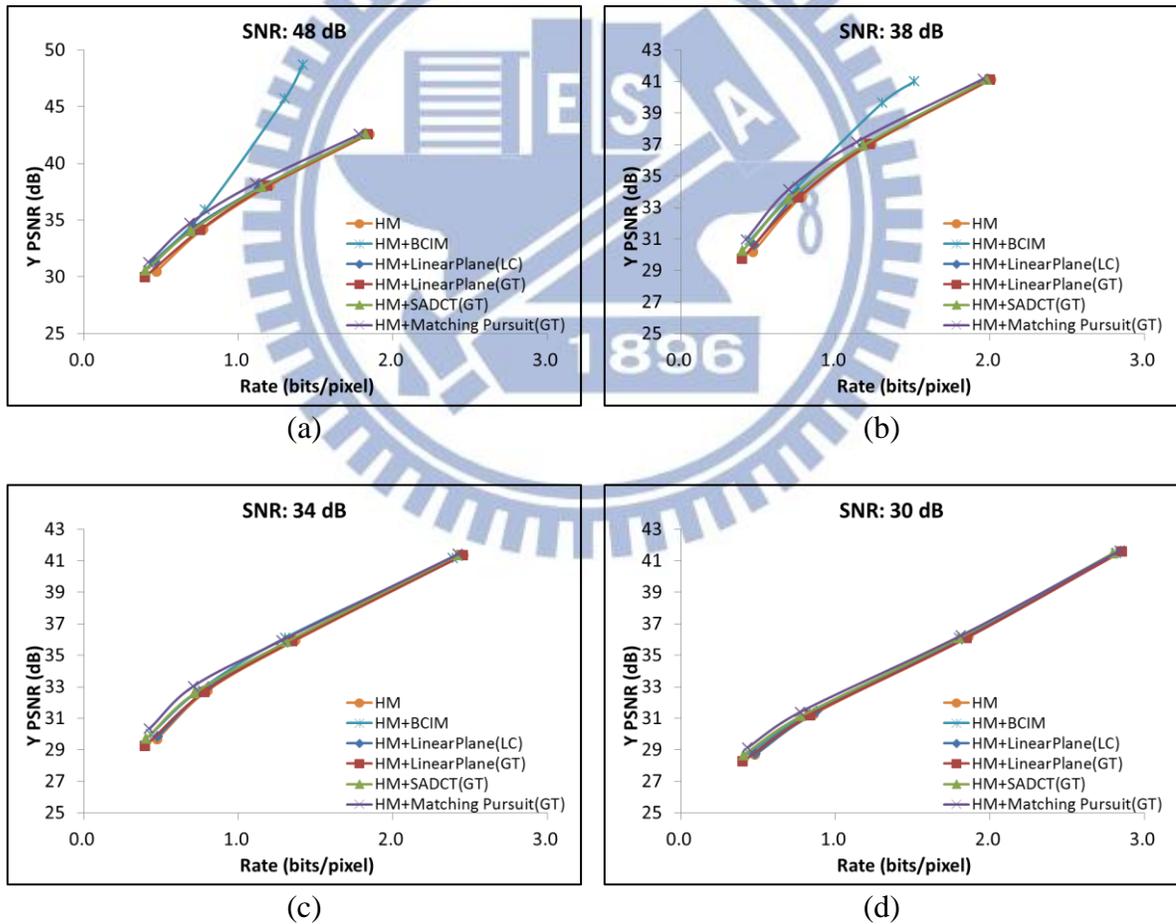


圖 59：Tile 在不同雜訊強度的 Rate distortion curves

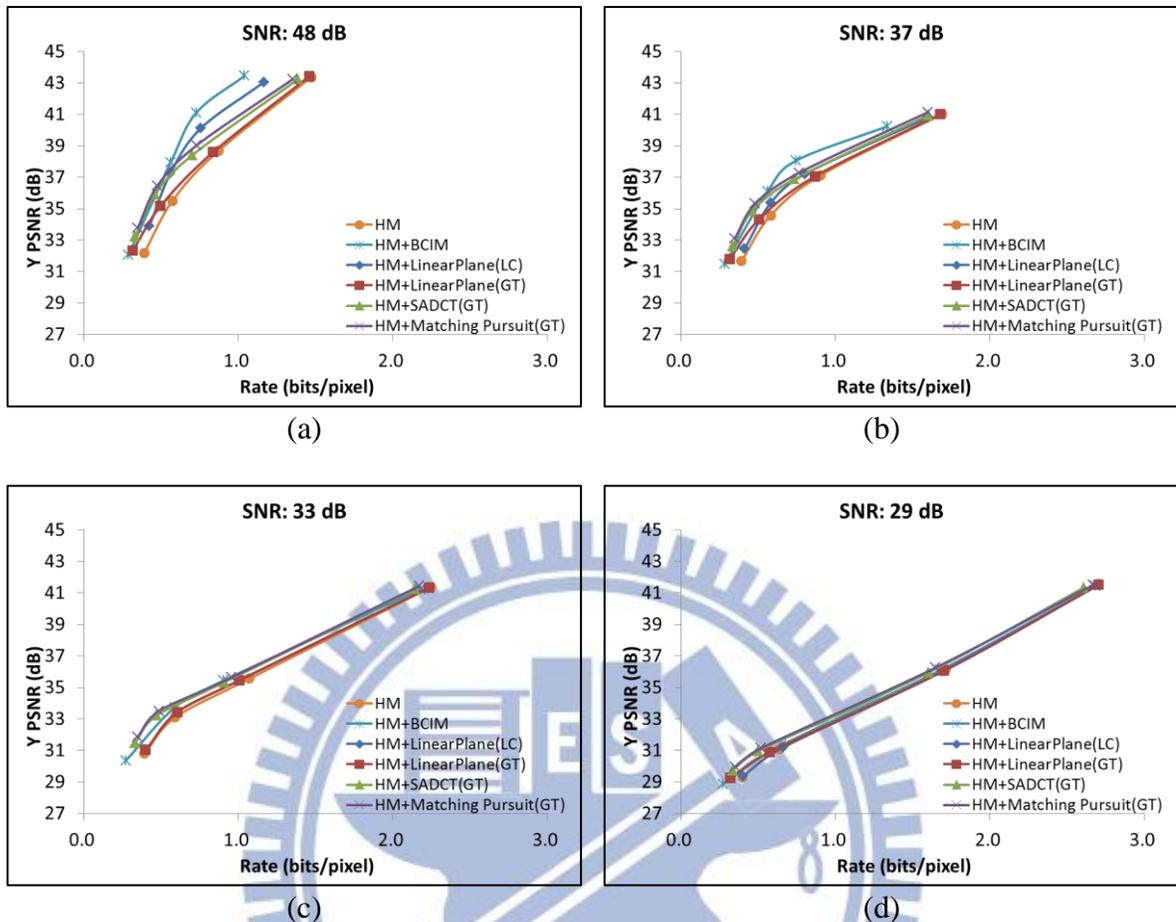


圖 60：Water waves 在不同雜訊強度的 Rate distortion curves

4.3 SADCT 的問題討論

從上述的實驗結果可以發現，自然背景採用 SADCT 的方式其壓縮效能，相較於 Matching Pursuit 其位元率的節省是較差的，而從圖 48 的主觀視覺相比，也可以發現使用 SADCT 會造成一些非常不自然的失真紋路，而 Matching Pursuit 的方法則不會有，因此以下將探討 SADCT 所遇到的問題。

不同頻率 AC 係數一起做水平 DCT

首先從 SADCT 的演算法中可以發現，當對每行像素做完垂直 DCT 後，便會對齊不同維度的垂直係數(Coefficient)，並根據其頻率順序，做水平方向的 DCT 轉換，如圖 61 左圖所示 DC 和 DC、AC1 和 AC1 等，但根據 $DCT(4)$ 的式子，當 $N = 8$ 和 $N = 4$ 時其 AC1 所對應的頻率不同，也就表示其 $DCT_N(n, k)$ 值不同，因此也就會造成將頻率不同的垂直係數，放在一起做水平方向的 DCT，因此在此例子下如果保持頻率相同的情況下做水平 DCT，應該如圖 61 右圖的配對，然而此例子是在 N 均為偶數的情況之下，如果 N 為奇數則仍無法找出其相同的頻率係數，從以上例子可以發現 SADCT 在對垂直係數做水平 DCT 時，其存在係數配對上的問題。

利用 Markov model 求出最佳的垂直係數配對

由於 SADCT 存在垂直係數(Coefficient)配對的問題，因此[23]使用 Markov model 以求出有最大相關性(Correlation)的垂直係數配對，從計算出的配對中可以發現，其最大相關性的垂直係數配對，並不是一定如圖 61 右圖或圖 61 左圖般配對，而是會依據其原始像素的位置及每行像素個數而有所變化，如圖 62 所示，當原始像素位置為圖 62 左圖時，則最大相關性為圖 62 左圖中線段的配對，而當原始像素位置為圖 62 右圖時，則配對改變，應如圖 62 中右圖線段的配對，因此為了得到更好的客觀壓縮效果，SADCT 中垂直係數的配對，應根據其原始像素間的位置及像素個數做配對。

Scaling factor 的影響

根據 Markov model 的模擬，SADCT 可以做最佳的垂直係數配對，以上方法雖可能提供更佳的客觀數據，但由於做垂直 DCT 時會因像素個數不同，而使用不同維度的 DCT，因此每行間的 Scaling factor 也會不相同，便會存在如圖 34 所造成的額外波形，因此 SADCT 的重建像素會產生如圖 48(c)般水平方向不連續的情形，雖然 MPEG4 修改了 DC 值的演算法，避免了以上問題，然而此種演算法只能大致解決了 DC 所遇到的問題，但在其他 AC 上仍存在相同問題無法解決，因而造成主觀視覺上水平像素不連續的失真現象。

結論

從以上討論中可以發現，雖然 SADCT 能在非矩形形狀下執行 DCT，而不需要進行補洞或執行 Matching Pursuit 這類耗時的運算，但 SADCT 仍存在上述問題，且以上問題在維度越大的 DCT 下更為明顯，因此在之前的實驗中可以發現 SADCT 所造成主觀上的問題，及客觀數據上不及 Matching Pursuit 的原因，然而在 MPEG4 由於其最大 DCT 維度只到 8，且原本目的為處理自然物體和自然物體間邊界的問題，因此每行間像素個數的變化較為和緩，不像因為是處理文字的背景而產生維度較大的變化，所以上述所造成的問題較小，也較難從主觀視覺上發現，如圖 63 所示當使用 MPEG4 壓縮文字背景時，仔細觀察下的確也會造成些微的不連續現象，然而在目前 HM 所支持的 DCT 維度可到 32，且本論文是用來處理文字背景的情況下，上述問題便會明顯浮現，因而造成主觀及客觀上的問題。

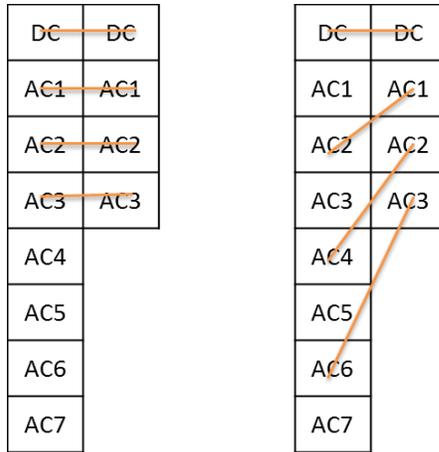


圖 61：SADCT 中橫向 DCT 的配對

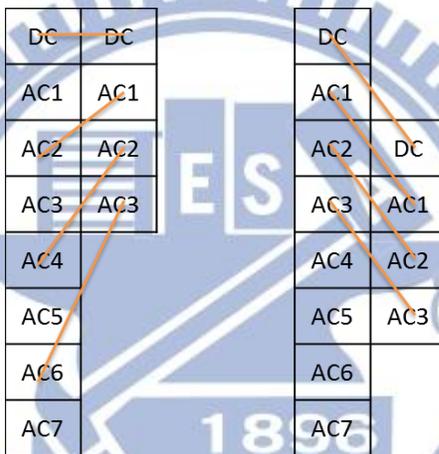
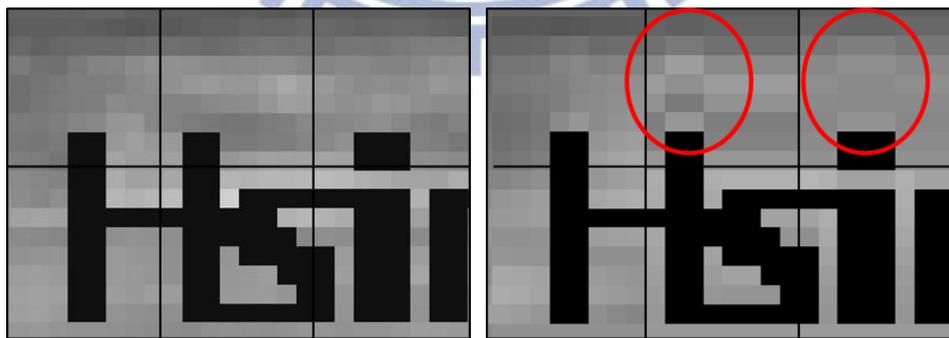


圖 62：DCT 係數其最大相關性之配對



(a) (b)

圖 63：MPEG4 壓縮結果(a) Original;(b) SADCT;

第五章、 結論

本篇論文首先分析了視窗影像特性及 Transform 所帶來的影響，並從分析中可以發現 DCT 會使得視窗影像中的區塊，造成能量分散的現象，不利於之後的壓縮。另外針對特定的視窗影像，提出了 BCIM 並結合特殊平面的編碼方式，此設計目的為將視窗影像分成圖像像素與文字像素，並找出圖像像素間其位於空間座標上的關係，並使用方程式表示成一特殊平面，本篇論文使用從只考慮區塊內其確定性關係的線性平面，以及根據統計上有相關性的 SADCT 來表達其此一特殊平面，從實驗結果可以看到當存在漸層時，使用線性平面便能有效表達其像素值的變化，而圖像像素為自然影像時，雖然利用 SADCT 處理有部分問題使得主觀及客觀效果不及 Matching Pursuit，但的確可以提升也可帶來客觀上的好處。

SADCT 相較於 Matching Pursuit 的方式在主觀及客觀上，SADCT 的表現較差，那是由於 SADCT 的確會在同一區塊內因為不同維度的 DCT 以及像素位置的移動，導致在進行橫向 DCT 轉換時造成一些問題，以致於在主觀及客觀上並無法達到有效的壓縮效果。

另外針對採用線性平面來表示此一特殊平面的方法，本篇論文提供一種藉由線性共識的切割方法，從實驗結果來看，這切割方法的確能切出有相關性像素間所產生的線性平面，並在大部分情況下達到和 Ground truth 相同的效果。

參考文獻

- [1] C. S. Boon, J. Takahashi, and S. Kadono, “Results of Core Experiments O4 and O9,” ISO/IEC MPEG 36th meeting, M1256, Chicago, Illinois, USA, September 1996
- [2] H. Nautsch and J. Ostermann, “Transform coding of compound images using Matching Pursuit,” *Proc. Picture Coding Symposium, PCS-2012*, Krakow, Poland, May 2012.
- [3] M. Mark and J. Z. Xu, “Improving screen content coding in HEVC by transform skipping,” *Proc. European Signal Processing Conference, EUSIPCO-2012*, Bucharest, Romania, August 2012.
- [4] J. Xu, G. J. Sullivan, and F. Wu, “Intra transform skipping,” ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC29/WG11 9th meeting, JCTVC-I0408, Geneva, CH, April 2012.
- [5] [Online]. Available: <http://www.realvnc.com/docs/rfbproto.pdf>
- [6] X. Li and S. Lei, “Block-based segmentation and adaptive coding for visually lossless compression of scanned documents,” *Proc. IEEE International Conference on Image Processing, ICIP-2001*, Thessaloniki, Greece, October 2001.
- [7] W. Ding, D. Liu, Y. He, and F. Wu, “Block-based fast compression for compound images,” *Proc. IEEE International Conference on Multimedia and Expo, ICME-2006*, Toronto, Canada, July 2006.
- [8] W. Ding, Y. Lu, and F. Wu, “Enable efficient compound image compression in H.264/AVC intra coding,” *Proc. IEEE International Conference on Image Processing, ICIP-2007*, San Antonio, USA, October 2007.
- [9] C. Lan, G. Shi, and F. Wu, “Compress compound images in H.264/MPGE-4 AVC by exploiting spatial correlation,” *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 946-957, April 2010.
- [10] Mixed Raster Content (MRC) ITU-T Study Group 8, Draft Recommendation T.44, March 1997.
- [11] R. L. de Queiroz, “Compressing compound documents,” in *The Document and Image Compression Handbook*, New York: Marcel-Dekker, 2005. (ISBN: 0-8493-3556-6)
- [12] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. L. Cun, “High quality document image compression using DjVu,” *Journal of Electronic Imaging*, vol. 7, no. 3, pp. 410–425, July 1998.

- [13]A. Zaghetto and R. L. de Queiroz, “MRC compression of compound documents using H.264/AVC-I,” *XXV Simpósio Brasileiro de Telecomunicações*, Recife, Brazil, September 2007.
- [14]H. Cheng and C. A. Bouman, “Document compression using rate-distortion optimized segmentation,” *Journal of Electronic Imaging*, vol. 10, no. 2, pp. 460–474, April 2001.
- [15]R. L. de Queiroz, “On data-filling algorithms for MRC layers,” *Proc. IEEE International Conference on Image Processing, ICIP-2000*, Vancouver, Canada, September 2000.
- [16]G. Lakhani and R. Subedi, “Optimal filling of FG/BG layers of compound document images,” *Proc. IEEE International Conference on Image Processing, ICIP-2006*, Atlanta, USA, October 2006.
- [17]S. Wang and T. Lin, “A Unified LZ and Hybrid Coding for Compound Image Partial-Lossless Compression,” *International Congress on Image and Signal Processing, CISP-2009 2nd*, Shanghai, China, October 2009.
- [18]S. Wang and T. Lin, “United coding for compound image compression,” *International Congress on Image and Signal Processing, CISP-2010 3rd*, Yantai, China, October 2010.
- [19]T. Lin, P. Zhang, S. Wang, K. Zhou, and X. Chen, “Mixed Chroma Sampling-Rate High Efficiency Video Coding for Full-Chroma Screen Content,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 173-185, January 2013.
- [20]T. Lin and P. Hao, “Compound image compression for real-time computer screen image transmission”, *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 993-1005, August 2005.
- [21]C. Lan, J. Xu, and F. Wu, “Compression of compound images by combining several strategies,” *IEEE International Workshop on Multimedia Signal Processing, MMSP-2011*, Hangzhou, China, October 2011.
- [22]F. Bossen, “Common HM test conditions and software reference configurations,” ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 9meeting, JCTVC-I1100, Geneva, CH, May 2012.
- [23]M. Bi, S. H. Ong, and Y. H. Ang, “Comment on Shape-Adaptive DCT for Generic Coding of Video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 6, pp. 686-688, December 1996.