

# 建構於 JPEG2000 及 EBCOT 壓縮技術中資料隱藏的研究 A New Scheme of Image Data Hiding Based on EBCOT and JPEG2000 Lossy Compression

陳同孝

鄭佳玉

Tung-Shou Chen

Chia-Yu Cheng

國立台中技術學院 資訊管理系

Department of Information Management, National Taichung Institute of Technology

台中市北區 404 三民路三段 129 號

129 Sec. 3, San-min Rd., Taichung, Taiwan 404, R.O.C.

tschen@ntit.edu.tw

nalunana@sinamail.com

## 摘要

JPEG2000 是目前最新的影像壓縮格式，JPEG2000 的最後一個步驟為 EBCOT (Embedded Block Coding with Optimized Truncation)，EBCOT 將熵編碼 (Entropy coding) 編碼完的位元流包裝成許多層，本文提出一個新的資料隱藏方法，將資料藏入於低頻位元流的最後一層中，並且利用層之較佳化的技巧，使低頻之最後一層的位元流長度增長，如此一來，可藏入的資料量也會提高。經實驗結果顯示，針對一張 512×512 像素的測試影像，在還原品質 34dB 之情況下，本方法可在影像中藏入之位元數均都高於 6700 位元以上，且藏完之後，其還原之影像品質是在人眼可接受範圍之內。

**關鍵字:** JPEG2000、資料隱藏、EBCOT

## Abstract

JPEG2000 is the newest compression technique for image. EBCOT (Embedded Block Coding with Optimized Truncation) is the last step in JPEG2000 lossy compression. After the generation of entropy coded bitstream, EBCOT separates the coded bitstream into many layers. In this paper, we propose a new scheme for data hiding based on EBCOT and JPEG2000. We embed data into the last layer of the low frequency bitstream. Moreover, we apply the technique called "optimized layer" to keep the higher embedding amount of the low frequency bitstream. It means that we can embed more data in an image. From experimental results, we see that, when the decoded image quality is limited to be better than 34dB, the proposed scheme can embed more than 6700 bits into an image. Besides, after the embedding, the decoded image quality is also acceptable for human visual.

**Keywords:** JPEG2000, Data hiding, EBCOT, Optimized layer

## 一、前言

資料隱藏將重要資料藏於偽裝影像中，而所謂重要資料可分為對此影像的著作權保護資料，或其它文字、影像、圖片等重要資料。偽裝影像通常為有意義的影像，資料隱藏藉著偽裝影像的外觀，來保護其重要資料，故人眼所見到的資料外觀，為有意義的偽裝影像，而誤以為此影像裡面沒有重要資料。

資料隱藏技術，可細分為二種研究方向，其中一種為「浮水印」的研究方向 [3][4][5]，另外一種為「大量資料藏入」的研究方向。在「浮水印」研究領域中，藏於影像的資料稱為浮水印資料，通常浮水印的資料量都很小，大約在幾百位元到 1 千多位元之間的資料量。而在「大量資料藏入」研究領域中，藏入的資料為大量資料，為有意義的資料。此領域要求的藏入資料量，希望可以愈多愈好，通常其資料量為浮水印資料量的好幾十倍，甚至百倍，且將資料藏於影像後，影像還原顯示時，都必須看起來與原始影像未藏有資料時，感覺都滿相同的，如 Wu 與 Tsai[8]所提出「資料藏入」之方法，在空間域影像中藏入大量的資料之後，此藏入之影像在還原時，都有符合剛剛所說的條件。在本文中，本文所提出「資料藏入」之方法，也都是屬於此「大量資料藏入」之研究領域的。

資料隱藏常利用影像壓縮技術，來壓縮偽裝影像，使偽裝影像體積變得更小，故此偽裝影像在傳輸過程中，所花費的傳輸時間比較少，傳輸過程也比較有效率。本文方法利用 JPEG2000 壓縮技術 [9]，來壓縮其偽裝影像。JPEG2000 在目前壓縮技術中，可算是最新的壓縮技術。JPG2000 即將成為國際標準壓縮格式，JPEG2000 和其它三種標準壓縮格式，JPEG、MPEG-4 VTC (Visual Texture Coding)、JPEG-LS，比起來，JPEG2000 比其

它格式擁有更多的優勢[2], JPEG2000 對於需壓縮的影像, 不但可以指定其更低的位元率, 且壓縮後的影像品質, 遠比其它三種格式的壓縮品質還要高出許多。在不久的將來, JPEG2000 即將取代過去 JPEG 壓縮技術, 成為網路上常用的影像壓縮格式。

在世界各地有許多團體和組織, 從事有關 JPEG2000 資料隱藏的研究, 然而這些成果大多為浮水印的研究, 沒有「大量資料藏入」之研究成果出現, 因此本文提出的「資料藏入」之方法, 可以算是創新的方法。關於 JPEG2000 浮水印研究的正式論文[3][4][5], 其實也不是很多。在這幾篇論文中, 提出的浮水印技術, 其成果都相當的輝煌, 也極具有實用的價值, 如在其中一篇的論文[5]中, 已藏入浮水印的影像, 可以抵抗 SPIHT 的失真壓縮, 且可以抵抗到 0.005bpp 的壓縮攻擊。可是浮水印研究和「大量資料藏入」為兩個完全不同角度的研究領域, 考量的因素也完全不一樣, 故如將浮水印技術[3][4][5]應用於「大量資料藏入」方面, 顯然較不恰當。而在這幾篇論文中, 所提出的浮水印方法, 都在頻率域中藏入浮水印資料, 若硬性地在 JPEG2000 之頻率域中藏入大量的資料, 其大量資料可能將無法完全地取回來, 因為在 JPEG2000 的壓縮過程中, 其中最後一道演算法為 EBCOT (Embedded Block Coding with Optimized Truncation)[6][7], EBCOT 將對頻率域係數, 進行編碼處理, 輸出一串的位元流, 接著 EBCOT 再對其中挑出不重要的位元群, 將其位元群丟棄, 來完成某種位元率的壓縮效果。而在頻率域中藏入大量的資料, 經 EBCOT 階段處理後, 其中有一部份的資料, 將會被偵測為不重要的位元群, 而遭 EBCOT 的丟棄, 故經過 EBCOT 處理後的影像, 其藏入的資料是無法完全地還原回來的。若想辦法在頻率域中, 計算某些的藏入位置, 使資料藏入此位置之後, 經過 EBCOT 階段處理, 藏入的資料仍不會遭 EBCOT 的丟棄。基本上, 這種方法在計算此資料藏入位置時, 仍需花費一段計算上的時間, 而且能藏入的資料量也極為有限, 同時其資料也無法百分之百地完全取回來, 故此類作法很難達到「大量資料藏入」的目的。

因此本文方法在 EBCOT 丟棄不重要的位元群之後, 才開始藏入大量的資料, 將資料藏入於 EBCOT 所產生的位元流中, 並滿足 EBCOT 位元流的規格, 如此藏入的資料不但具有隱蔽性, 且藏入的資料皆能百分之百地完全取回來, 由於取出的資料完全沒有任何的遺失, 故本文方法很容易達到「大量資料藏入」的目的。因此本文方法在「大量

資料藏入」領域方面, 和浮水印技術[3][4][5]應用於此領域方面比起來, 本文方法將較為優秀許多, 同時, 本文方法和 Wu 與 Tsai[8]提出的「資料藏入」之方法比起來, Wu 與 Tsai 之方法雖然可以在空間域中藏入大量的資料, 但比較沒有考慮到此藏入資料之隱蔽性, 而且被藏入之影像是不能透過某種壓縮技術, 來壓縮此影像的, 也就是說如果將資料藏於某張數位灰階影像中, 則此數位灰階影像之整體位元率必須為 8bpp, 所以基於此, 本文提出之方法是比較佔有優勢的。

本文方法對於藏入的資料, 沒有作任何前置的處理, 如將資料重新編碼等, 使藏入的資料體積變小。在本文中, 除了提出在 JPEG2000 之 EBCOT 中大量地藏入資料的方法之外, 同時亦研究探討將資料覆蓋於位元流中, 資料總共可以覆蓋多少個位元, 且被覆蓋後的影像還原時, 其還原影像品質是否能在人眼所接受的破壞範圍之內。在影像品質能被人眼接受的前提下, 本文將尋找總共可以覆蓋 EBCOT 位元流的最多位元數。同時, 本文方法亦將尋找此影像在低位元率下, 可藏入的最多位元數。而且本文也將分析, 低位元率和可藏入最多位元數之間的互動關係。當然, 此藏入資料如有經過壓縮, 然後再藏入 EBCOT 位元流的話, 則此資料藏入的量亦會提高數倍以上, 也就是說此真正的資料量為此張影像中, 可藏入最多位元數的數倍以上。第貳節將介紹 JPEG2000[9]壓縮流程和 EBCOT 優點, 第參節提出本文的方法, 第肆節說明實驗結果, 第伍節為總結論。

## 二、 JPEG2000 的影像壓縮技術

本文提出的 JPEG2000 壓縮技術, 遵循著 FCD Part1 (JPEG2000 Part1 Final Committee Draft version 1.0)[9]標準格式。在 JPEG2000 壓縮過程中, EBCOT 包含了熵編碼(Entropy coding)和位元率配置演算法(Rate allocation algorithm)。

### 2.1. JPEG2000 壓縮流程

JPEG2000 先利用離散小波轉換, 把空間域影像轉換為以頻率域來表示, 一般內定值為五次的小波轉換[9], 故本文之後所提到的小波轉換, 亦皆指五次的小波轉換。然後 JPEG2000 再經過量化的階段, 量化完之後, 才開始進入熵編碼(Entropy coding)的過程。JPEG2000 將每個頻帶分割為許多不重疊的矩形區塊, 同時, 每個區塊都將獨立地進行熵編碼的流程, 故每個矩形區塊又稱為編碼

區塊(Codeblock),通常編碼區塊為 $64 \times 64$ 的區塊[9]。在每個編碼區塊中,都有許多的位元平面,熵編碼先從高位元平面(即重要位元平面)開始編碼,然後再換下一個較不重要的位元平面,同時,每個位元平面都會進行三個 passes 來編碼,此三個 passes 為三個不同的編碼規則,且配合著算術編碼一起來編碼,所以每個 passes 都會送出一串編碼後的位元流。然後每個 pass 所產生的位元流,再依照每個 pass 編碼的前後順序來排列,也就是說,最先編碼的位元流排在最前面。之後,JPEG2000 再將熵編碼編碼完的位元流,送給位元率配置演算法。

對每個編碼區塊的位元流而言,位元率配置演算法在每個 pass 所產生的位元流後面,假想成有一個切斷點(Truncation point),故此編碼區塊有很多個切斷點,然後再由 PCRD (Post-Compression Rate-Distortion optimization algorithm)來決定這些切斷點中,哪些為有效的切斷點,如果在這些切斷點中,有某個切斷點被 PCRD 決定為有效切斷點的話,則此切斷點將會有一個貢獻程度(Slope)。假設此編碼區塊有 $n$ 個有效切斷點,則針對每個 $n$ 而言,其貢獻程度以 $S_n$ 來表示之, $S_n$ 為 $\Delta D_n / \Delta R_n$ 。 $\Delta R_n$ 為位元流長度,此長度為此 $n$ 和前一個有效切斷點之間的位元流長度。 $\Delta D_n$ 為品質測量程度,PCRD 將計算,當此編碼區塊接收到 $\Delta R_n$ 的位元流時,利用此位元流來還原此編碼區塊的係數,則此編碼區塊的還原品質,將比之前還沒有收到此位元流的還原品質,還要好一點點,則其好一點點的差距,即為 $\Delta D_n$ 。之後,本文若有再提到切斷點的名詞,此名詞皆代表為有效切斷點。

每個編碼區塊經過 PCRD 處理之後,位元率配置演算法將全部編碼區塊的位元流,包裝成好幾個輸出層(Layer)來輸出。每個輸出層將分別從每個編碼區塊中,取此編碼區塊中的一小部份位元流,故此輸出層將會有從全部編碼區塊中,各取其編碼區塊中一小部份的位元流。在此輸出層中,每個一小部份的位元流稱為「貢獻」(Contribution),則此一小部份位元流的長度,稱為「貢獻長度」。比如說,假設全部總共有 5 個編碼區塊,則此輸出層將從 5 個編碼區塊中,各取其編碼區塊中的一小部份位元流,則此輸出層總共有 5 個一小部份位元流,故此輸出層總共有 5 個貢獻,假設其中某一個貢獻的長度為 30 位元組,則此 30 位元組稱為此貢獻的「貢獻長度」。

同時,此輸出層也會有自己的層門檻值(Threshold),而此層門檻值是來決定,此輸

出層可以取到多少個貢獻。比如說,在此輸出層中,每個貢獻的取法,為從某個編碼區塊中的某個切斷點開始取,然後直接取到此編碼區塊中的另一個切斷點為止,而此另一個切斷點的 $S_n$ ,都必須大於此輸出層的層門檻值。如果此 $S_n$ 大於此層門檻值的話,則此輸出層才可以取此編碼區塊中,此兩個切斷點之間的位元流。故此層門檻值只是一種衡量的標準而已,來決定此兩個切斷點之間的位元流,是否要被取進來,而成為此輸出層的貢獻。然而此層門檻值並沒有什麼特別的意義,所以此層門檻值的值域範圍和 $S_n$ 很類似。

在每個輸出層中,每個貢獻長度的長短,決定於此輸出層的層門檻值。此層門檻值若比較小,使某個編碼區塊的 $S_n$ ,大於此層門檻值的機率變多,故此輸出層很容易取到此編碼區塊的位元流,同理,此輸出層將容易從其它全部編碼區塊中取到其編碼區塊的位元流。因此,此輸出層所取到的貢獻數目將會增多,且其中每個貢獻的「貢獻長度」,也將會增長許多。反之,此層門檻值若比較大,使每個編碼區塊的 $S_n$ ,大於此層門檻值的機率變少,則此輸出層所取到的貢獻數目,將會變少,且其中每個貢獻的「貢獻長度」,也會減少許多。

當位元率配置演算法的整個流程結束之後,如果在某個編碼區塊中,還有剩餘的位元流,沒有被所有輸出層所取到,則此位元流將會遭到位元率配置演算法的丟棄。故位元率配置演算法為了滿足某種位元率的壓縮效果,來大量地丟棄這些剩餘的位元流。

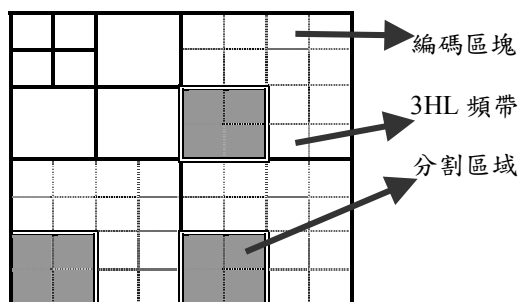
## 2.2. 視覺遮罩

當每個影像還原時,都會通過 JPEG2000 之解碼端的視覺遮罩階段(Visual masking)[6][7]。視覺遮罩對於空間域中能量變化比較大的某塊區域而言,此區域通常會有好幾個邊緣(Edge)經過,則視覺遮罩對於此有邊緣經過的地方,將會修飾的較清楚,所以在此區域中有邊緣經過的地方,將會變的較清晰。反之,對於空間域中能量變化比較平緩的地方,如平滑的區域而言,視覺遮罩對於此區域將會修飾的比較少,故此區域在影像還原時,比較不會受到視覺遮罩修飾的影響。然而本文方法將資料藏於影像後,其影像還原時,將會受到視覺遮罩修飾的影響,至於其影響的程度,本文將在 3.1.1 節中作個說明。

## 2.3. EBCOT 之優點

EBCOT 將許多的頻帶，依照頻率高低的不同，來分成許多不同的解析度 (Resolution)，以  $R_r$  來表示其解析度， $R_r$  的值域範圍為  $\{R_0, R_1, R_2, \dots, R_6\}$ ， $R_0$  為最低的解析度，如為最低頻率的 5LL 頻帶，其餘的解析度  $R_r$ ，其值域範圍為  $\{(6-r)LH, (6-r)HL, (6-r)HH | 0 < r \leq 5\}$ ，如  $R_1$  為較低頻率的 5LH、5HL、5HH 頻帶， $R_2$  為次低頻率的 4LH、4HL、4HH 頻帶，其它較大的解析度，將依此公式如此推算。

在每個  $R_r$  裡面所包含的頻帶中，如果其中有幾個編碼區塊，都對應到相同的空間域位置，則這幾個編碼區塊視為一個分割區域 (Partition position)[1][9]，如圖一所示，圖一為  $R_3$  所包含的頻帶圖，圖中三個灰黑色部份所指的編碼區塊，都對應到相同的空間域位置，則這幾個編碼區塊都視為一個分割區域。然後在每個輸出層中，EBCOT 將同個分割區域的貢獻，包裝成一個封包(Packet)，故每個輸出層將會有很多個封包。



圖一：在  $R_3$  中某個分割區域圖

JPEG2000 因實作 EBCOT，而擁有許多的優點，如隨機存取(Random access)[6][7]、SNR 漸近式傳輸 (Signal to Noise Ratio scalability)[1][7]、空間域漸近式傳輸 (Spatial scalability)[1][7]。隨機存取應用於，只想還原空間域中某個區域，而其他的區域可以不必還原時。隨機存取將取所有輸出層中可以還原成此區域的相關封包，然後利用此相關封包，來還原此區域的影像內容。SNR 漸近式傳輸指，在解碼端時，所還原的影像品質，可以漸漸的愈變愈清楚。SNR 漸近式傳輸利用比較早輸出的輸出層，來還原此空間域影像中的基本輪廓，然後較後面輸出的輸出層，來加強此影像基本輪廓的品質。在空間域漸近式傳輸方面，編碼端先將位元流依  $R_r$  大小，由小依序的排到大 ( $R_0$  為解析度最小的， $R_5$  為解析度最大的)，然後先傳輸解析度較小的位元流，再依序傳輸解析度較大的位元流。則解碼端先接收解析度較小的位元

流，來還原成比較小的空間域影像，然後再接收解析度較大的位元流，來還原成比較大的空間域影像。以上所述之 EBCOT 優點，功能都非常的強大[2]，所以本文將在 3.3 節中，將此文提出「資料藏入」之方法和這些優點之間的關係，作個說明。

### 三、 資料隱藏法

#### 3.1. 資料的藏匿

EBCOT 之最後一層有許多的封包，其中有關  $R_0$  的封包，本文稱之為  $R_0$  封包。同理，其中有關  $R_1$  的封包，本文稱之為  $R_1$  封包。其中有關  $R_2$  的封包，本文稱之為  $R_2$  封包。而本文提出的「資料藏入」之方法，將資料藏入於此  $R_0$ 、 $R_1$ 、 $R_2$  封包裡面，將資料直接覆蓋於這些封包裡面的位元流。3.1.1 節說明為何將資料藏入於  $R_0$ 、 $R_1$ 、 $R_2$  封包裡面，3.1.2 節說明如何使藏入的資料量變多。

##### 3.1.1. 藏入之位置

編碼區塊可分為高頻的編碼區塊和低頻的編碼區塊，其中高頻的編碼區塊為，從高頻頻帶中取出的編碼區塊，反之，低頻的編碼區塊為從低頻頻帶中取出的編碼區塊。低頻編碼區塊的位元流，比高頻編碼區塊的位元流還要來的重要，故其低頻編碼區塊的  $S_n$  值，比高頻編碼區塊的  $S_n$  值還要高出許多。以  $512 \times 512$  Lena 影像為例，實驗結果如表一所示，實驗的程式碼(Source code)為 JJ2000 VM4.1(Verification Model)[10][11] 的程式碼，1、2、3 分別代表為第 1 個、第 2 個、第 3 個切斷點，5LL 和 5HL 代表低頻的編碼區塊， $B_1$ 、 $B_2$  代表高頻的編碼區塊，其高頻編碼區塊為從 1HH 頻帶中取得的。

表一：不同高低頻編碼區塊的  $S_n$  值

編碼 區塊	Lena		
	1	2	3
5LL	137.59	105.48	62.67
5HL	22.44	22.16	14.30
$B_1$	0.0013	0.0004	0.0003
$B_2$	0.0003	0.0002	0.0001

高頻編碼區塊的  $S_n$  值比低頻編碼區塊的  $S_n$  值還要低許多，故愈前面輸出層所包含的貢獻，幾乎都是從低頻編碼區塊中取出的位元流，很少有從高頻編碼區塊中取出的位元流。以  $512 \times 512$  Lena 影像為例子，以兩個

編碼區塊來實驗，其中一個編碼區塊為從 5LL 頻帶中取得的，此編碼區塊代表低頻的編碼區塊。另外一個編碼區塊為從 2HH 頻帶中取得的，其編碼區塊代表高頻的編碼區塊，以  $CB$  來表示之。在 EBCOT 之每個輸出層中，將會有此兩個編碼區塊的貢獻，將每個輸出層中的此兩個貢獻長度，作個一覽表，如表二所示之。層編號代表每個輸出層的編號，其編號值愈小，代表此輸出層愈早輸出，在 J2000 VM4.1 的程式碼中，其貢獻長度的單位，以位元組(Bytes)為基本單位 [11]。表二的數據將可發現在第 0 層到第 9 層時，仍然沒有高頻編碼區塊的貢獻，直到第 10 層時才有高頻編碼區塊的貢獻。在此例子中，第 10 層為最後一層，如果將資料分別覆蓋於第 10 層中的高低頻貢獻後，對這張影像而言，將產生不一樣的破壞，而將資料覆蓋於第 10 層之低頻貢獻後，所產生的破壞是比較小的，因此本文提出的方法，決定將資料覆蓋於 EBCOT 之最後一層的低頻貢獻中，而  $R_0$ 、 $R_1$ 、 $R_2$  封包裡面的位元流，乃屬於低頻的位元流。

表二：貢獻長度變化表(單位：位元組)

層編號	0	1	2	3	4	5	6	7	8	9	10
5LL	100	29	0	26	8	0	0	31	0	0	33
$CB$	0	0	0	0	0	0	0	0	0	0	11



圖二：部份還原影像

圖二為將資料覆蓋於第 10 層的高頻貢獻後，再還原的部分影像。由此圖可發現，將資料藏入此張影像後，將產生一些破壞，而此影像經過 JPEG2000 之視覺遮罩修飾之後，使原本的破壞更趨於明顯。而造成此現象的原因在於，視覺遮罩對於空間域中能量變化比較大的區域，將會修飾的更多，而圖二被藏入的貢獻，屬於高頻頻率的貢獻，其貢獻相當於空間域影像中高頻頻率的能量，故原本的破壞經視覺遮罩修飾之後，彷彿呈現粒狀波形的感覺。本文經過不斷的實驗後，發現將資料藏入於最後一層的  $R_0$ 、 $R_1$ 、 $R_2$  之封包裡面，比較不會受到視覺遮罩修飾的影響。

### 3.1.2. 層之較佳化(Layer Optimization)

位元率配置演算法將產生許多的輸出層，以  $Q$  來表示所有的輸出層， $Q$  的內容為  $\{Q_0, Q_1, \dots, Q_{q-2}, Q_{q-1}, Q_q\}$ ，其中  $Q_0$  代表第 0 層， $Q_{q-1}$  代表倒數第二層， $Q_q$  代表最後一層。EBCOT 對於第  $Q_{q-1}$  層，可以指定其輸出層的位元率，以  $br$  來表示之， $br$  必須小於整體位元率值，指定  $br$  值的動作，又稱為對第  $Q_{q-1}$  層作「層較佳化」的動作 [10][11]，而本文提出的「資料藏入」之方法，利用「層較佳化」之技巧，將可使藏入的資料量變很高。

假設對一張  $512 \times 512$  之  $M$  影像而言，當本文方法對於其作「層較佳化」的動作時，首先對第  $Q_{q-1}$  層下一個  $br$  值，而此  $br$  值即為之前所說的  $br$ ，目的是為了使現在所有的輸出層，再重新作一次分配輸出層之動作，也就是說現在所有輸出層之內容，將會作一次重新的調整。而此下  $br$  的動作，只不過是個虛擬的動作，只是標記著第  $Q_{q-1}$  層有一個位元率。接著，再將頻率域中所有編碼區塊的位元流，再重新送給位元率配置演算法，當位元率配置演算法接收到這些所有位元流之後，開始將這些所有位元流包裝成許多的輸出層。但因受限於第  $Q_{q-1}$  層之  $br$  因素，所以位元率配置演算法會先計算第 0 層到第  $Q_{q-1}$  層之間的位元流長度，使之長度剛好等於  $len_{br}$  值， $len_{br}$  如公式(1)所示。 $W$  為此張影像的寬度， $H$  為此張影像的長度， $Header$  為此張影像之固定檔頭長度(經過 JPEG2000 壓縮後的檔案，其檔頭長度永遠都是固定的)，因  $Header$  單位為位元組，所以要將  $W \times H \times br$  的單位由位元轉為位元組。例如，假設  $br$  為  $0.015\text{bpp}$ ，影像長寬為  $512 \times 512$ ，固定檔頭長度為 129 位元組，則  $len_{br}$  為  $512 \times 512 \times 0.015 / 8 - 129 = 362.52$  位元組。

$$len_{br} = W \times H \times br / 8 - Header \quad (1)$$

同時，位元率配置演算法也會計算每個輸出層之預估長度，計算完之後，每個輸出層開始取出編碼區塊中的位元流，將取出的位元流裝進自己輸出層中，而成為貢獻。然而每個編碼區塊的切斷點是經 PCRD 決定的，故在每個編碼區塊中，切斷點和切斷點之間的位元流長度，也是經 PCRD 決定的，所以任何兩個切斷點之間的位元流長度，對於輸出層而言，都是固定的，輸出層是沒有

權利可以改變此位元流長度的，並且每個輸出層都要記載著自己輸出層之一些額外的資訊，如自己輸出層中有多少個貢獻，且每個貢獻長度又為多長等等。因此光這些如上所述之因素，就足以產生每個輸出層實際上所取出的貢獻總長度，是沒有辦法剛好等於其輸出層的預估長度，因此就會產生此輸出層實際上所取出的長度和其預估長度之間，有誤差的情況發生。

因每個輸出層都會有微小的誤差，然而當位元率配置演算法一直包裝到第  $Q_{q-2}$  層時，此一直累積的誤差將會產生更大的誤差，而幸好地，本文提出「資料藏入」之方法，有對第  $Q_{q-1}$  層下  $br$  值，故位元率配置演算法將會控制第  $Q_{q-1}$  層所取出的位元流長度，使第  $Q_{q-1}$  層之位元流長度，再加上第 0 層到第  $Q_{q-2}$  層之間的位元流長度後，將會非常接近於  $len_{br}$  值。因此，位元率配置演算法就會對第  $Q_{q-1}$  層之層門檻值，作上下之間的變動調整，找到一個最適當的層門檻值，使此層門檻值將可以使第  $Q_{q-1}$  層所取到的位元流長度，剛好可以符合所要求的長度。在 JJ2000 VM4.1 的實作中，JJ2000 VM4.1 透過不斷的模擬(Simulation)，來找到第  $Q_{q-1}$  層之層門檻值的最佳值，其模擬的方式類似最佳化逼近法之方式，有興趣的讀者可以在相關文獻[10][11]中找到。

基於上述之觀念，本文提出「資料藏入」之方法，利用「層較佳化」之技巧，藉著  $br$  值來控制第 0 層到第  $Q_{q-1}$  層之間的位元流長度，自然而然也就可以使第 0 層到第  $Q_{q-1}$  層之間的位元流長度縮小，當然在  $M$  之整體位元率完全沒有被更改的情況下，位元率配置演算法需將取更多的位元流，來補第  $Q_q$  層的位元流空間，使第 0 層到第  $Q_q$  層之間的位元流長度，剛好等於其整體位元率下的壓縮效果，因此最後一層之  $R_0$ 、 $R_1$ 、 $R_2$  的位元流長度也會跟著增長，所以可藏入的資料量也會跟著變許多。

本文方法對於藏入的資料量，將採變動的方式，來調整其資料量的多寡，藏入的資料量大小沒有固定，其資料量和  $br$  值之間，將產生互動的關係。基本上， $br$  值和藏入資料量之間，將成反方向的變化。當藏入的資料量逐漸地變很大， $br$  值將呈反方向逐漸地變很小，當藏入的資料量逐漸地變很小， $br$  值將呈反方向逐漸地變很大，故本文方法的可行性相當高。基於上述之觀念後，如果在某張影像的整體位元率固定之情況下，而想

要尋找此張影像的最多藏入資料量時，則  $br$  值必須要變得很小，同時此藏入之影像也必須在人眼可接受範圍之內，則此藏入資料量才可稱為「最多藏入資料量」。至於此  $br$  值和「最多藏入資料量」之間的關係，本文有在第肆節之實驗結果中，加以分析和討論。

### 3.2. 資料之取出

當解碼端接收到此藏入之影像時，本文方法需在此影像之整個解碼動作之前，先取出其所藏入之資料。首先解碼端會先解析每個輸出層之封包，也就是先解析每個輸出層之一些額外的記載資訊，因此，本文方法可以透過此解析過程，就可以輕易地知道，哪一個封包是我們所藏入之封包，自然而然地，本文方法也就可以順利地取出最後一層之  $R_0$ 、 $R_1$ 、 $R_2$  封包，然後再打開這些封包，直接取出此封包裡面的位元流。並且，取出之位元流必須透過原先在編碼端之藏入順序，然後按照其藏入之順序，一一地被銜接起來。銜接完後之位元流，即為原本所藏入之資料。

### 3.3. 藏入之分析

在 EBCOT 之 SNR 漸近式傳輸方面，最後一層之  $R_0$ 、 $R_1$ 、 $R_2$  封包，將會隨著最後一層之位元流，來傳送到解碼端。而在解碼端方面，解碼端是在逐漸還原的最後階段中，才開始還原此最後一層之位元流的，故人眼也是在逐漸還原的最後階段中，才開始看見最後一層之位元流的還原影像。因此，在此逐漸還原的過程中，人眼看到此逐漸還原的還原影像，和原本正確的還原影像之間，感覺都滿相同的。

在 EBCOT 之空間域漸近式傳輸的還原過程中，剛開始  $R_0$  位元流會先傳送到解碼端，而在此位元流中，有參雜著我們所藏入之資料流，所以在解碼端時，解碼端先利用此位元流所還原的小空間域影像，可能會有一點點的小破壞，然而這種破壞都在人眼可接受範圍之內，故人眼是較看不太出來的。同樣的，解碼端陸續地接收不同解析度的位元流，而漸近地還原出不同大小的空間域影像，然而，對於這些漸近還原的影像，也全部都在人眼可接受範圍之內。

在 EBCOT 之隨機存取方面，如果在極端之情況下，如對很大的影像進行壓縮時，因其影像之長寬很大，故此影像經 JPEG2000 之小波轉換後，在  $R_0$ 、 $R_1$ 、 $R_2$  所包含的頻帶中，以  $S$  來表示這些頻帶所包含編碼區塊之全部集合。在小波轉換之頻帶中，因受到



此張影像之長寬很大的影響，故每個頻帶之長寬也會變得很大，因此，此頻帶所包含之編碼區塊將會有許多個。而小波轉換本身就具有空間域對應關係，故每個頻帶中的每個編碼區塊也都具有空間域對應關係，由此可知，如果將資料覆蓋於  $S$  中某個編碼區塊之位元流時，此覆蓋後的結果，就相當於對空間域中某個區域進行破壞。然而，本文提出「資料藏入」之方法，完全不受此因素之影響，因為本文方法有善加利用「層較佳化」之技巧，藉著第  $Q_{q-1}$  層之  $br$  的縮小，將可使第  $Q_q$  層完全取到  $S$  中所有編碼區塊之位元流，故第  $Q_q$  層之  $R_0$ 、 $R_1$ 、 $R_2$  封包將包含  $S$  中所有編碼區塊之位元流，而本文方法所藏入之封包，即為這些的封包，故本文方法將資料藏入後，其所引起的破壞對於空間域影像而言，為整體性的破壞，而非區域性的破壞，自然而然地，此還原影像也就沒有區域和區域之間，有類似區塊效應的情形發生。因此大眾在存取此張影像的某塊區域時，也完全都不會發現此塊區域有任何的異狀，自然而然地，大眾也難以從其區域來發覺此張影像有藏入過資料。

#### 四、 實驗結果

本篇論文所使用之實驗平台為 Pentium III 733MHZ CPU、128MB 記憶體、MS Windows 2000 Professional 之作業系統、Adobe Photoshpe5.0 之數位影像顯示工具，實驗的程式碼為 JJ2000 VM4.1[10][11]之程式碼，以 PSNR 值之評估方式來評估影像品質之好壞。

表三：臨界區間範圍表

	PSNR 約為 30.77		PSNR 約為 34	
	位元率 (bpp)	PSNR (dB)	位元率 (bpp)	PSNR (dB)
Lena	0.1	30.74	0.2	34.29
Girl	0.12	30.84	0.25	34.17
Gold	0.17	30.77	0.4	34.09
Barbara	0.35	30.76	0.6	34.01

本文以四張  $512 \times 512$  且不同性質之數位灰階影像，如 Lena、Girl、Gold、Barbara 來作實驗，在這些測試影像中，Lena 屬於比較平滑的影像，而 Barbara 屬於比較複雜的影像。在本文之實驗中，每個測試影像都有自己的低位元率之實驗範圍，也就是在此位元率範圍中來作實驗，而此位元率之實驗範圍，本文稱之為臨界區間範圍。臨界區間範圍的起始值和終止值，沒有一定的限制，在

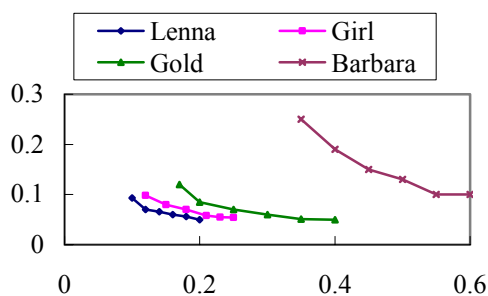
本文中，每個測試影像之臨界區間範圍如下：Lena 為 0.1bpp 到 0.2bpp，Girl 為 0.12bpp 到 0.25bpp，Gold 為 0.17bpp 到 0.4bpp，Barbara 為 0.35bpp 到 0.6bpp，如表三所示之。

對於某張影像而言，如果要尋找此張影像之「最多藏入資料量」，則必須考慮到許多的影響之因素，而這些因素如果控制不好的話，都將會造成實驗數據上的誤差，這些因素分別如：此張影像之複雜程度、所藏入之資料流的特性、此張影像的整體位元率為何、和此藏入之影像可被人眼接受時，而能接受到怎樣的程度。因此，本文為了使之後的實驗數據更趨於準確，將選固定之資料流來作為藏入之資料流，而不被不同之資料流所產生 PSNR 值上之誤差所影響。並且，本文定義此藏入之影像，能被人眼所接受時，其藏入之影像品質至少都要大於 30dB，且都在 30dB 以上。

當之前所說的影響之因素控制好之後，本文開始探討在某張測試影像之下，在某個整體位元率的壓縮下，來尋找此張影像之  $br$  值為何、和此「最多藏入資料量」為多少，尋找完之後，其  $br$  值、和此「最多藏入資料量」之實驗數據如表四所示。表中位元率代表此張影像之整體位元率，表中資料量代表此張影像之「最多藏入資料量」，表中百分比代表此「最多藏入資料量」佔此張影像之檔案大小的比例。如對 Lena 而言，在整體位元率為 0.2bpp 之下，如果  $br$  值為 0.05bpp，則 Lena 將可藏入最多資料量為 6768 位元，且其資料量佔 Lena 之檔案大小約為 12.91%，並且，此藏入之 Lena 的還原影像品質都在 30dB 以上。在此例子中， $br$  值絕對不能小於 0.05bpp，因為一旦小於 0.05bpp，雖然此張影像之可藏入資料量又變的更高，然而這張影像之還原品質破壞又變的更嚴重，所以此張影像之還原品質已經沒有在人眼可接受範圍之內，故此張影像的  $br$  值必須為 0.05bpp，則此張影像之藏入資料量才為「最多藏入資料量」，也就是為 6768 位元。

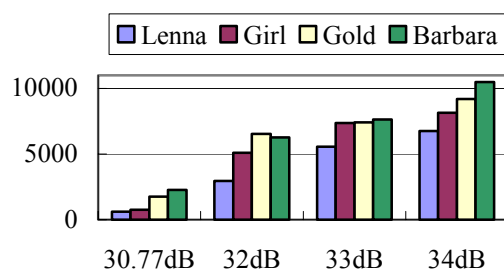
如果將此張測試影像在臨界區間裡，其整體位元率值和  $br$  值之間的互動關係，可以畫成如圖三來表示。由圖三的曲線中，可以發現當此張測試影像在整體位元率值逐漸地變很大時， $br$  值都會跟著變很小，這種發現說明了，因整體位元率值的增高，和  $br$  值的縮小，使第  $Q_q$  層之  $R_0$ 、 $R_1$ 、 $R_2$  的位元流長度，也會跟著變很大，而且是雙重的增長，自然而然地，此張影像之「最多藏入資料量」，也會跟著雙重提高。然而在表四中，整體位元率值如果從某個位元率值，非常輕微地移動到另外一個位元率值，對於最後一層

之  $R_0$ 、 $R_1$ 、 $R_2$  的貢獻而言，僅管第  $Q_{q-1}$  層和第  $Q_q$  層之層門檻值，不斷地作上下之間的變動調整，來找到最佳化之層門檻值，然而有時最後一層之  $R_0$ 、 $R_1$ 、 $R_2$  的貢獻長度，仍然沒有什麼明顯地增長。如表四中 Lena 的整體位元率值，從 0.18bpp 輕微地移動到 0.2bpp，表四之 Lena 的百分比，卻從 13.2% 輕微地降到 12.9%。此百分比的變化，很容易使人誤解為此張影像之「最多藏入資料量」，都沒有什麼明顯地增加，其實不然，此 12.9% 只是代表 Lena 在 0.2bpp 下，「最多藏入資料量」佔其檔案大小，比較小一點點罷了。



圖三：位元率值和  $br$  值之曲線圖，橫軸為位元率值(單位：bpp)，縱軸為  $br$  值(單位：bpp)

圖四為此四個測試影像之「最多藏入資料量」的比較圖，其目的是要比較，本文提出「資料藏入」之方法是較適用於哪一種性質之測試影像。在此圖中，是先將四個不同之測試影像，在未藏入資料時，先還原成相同影像品質之後，此時，此影像品質之評估方式，採用 PSNR 值之評估方式，然後再比較此四張影像之「最多藏入資料量」。如 Lena 在未藏入資料時，可以在整體位元率為 0.2bpp 下，還原成相同影像品質約為 34dB，然後，此 Lena 之「最多藏入資料量」可以到達 6768 位元。



圖四：測試影像之可藏入最多資料量比較圖，橫軸為藏入前的 PSNR 值(單位：dB)，縱軸為可藏入的最多資料量(單位：bits)

由圖四之圖表中，可以發現愈複雜之影

像，在相同影像品質下，其「最多藏入資料量」，是比其它較不複雜之影像還要多的。然而，在一般人的觀念中，都會以為藏入愈多之資料，此藏入之破壞相對的也就愈大，因此，圖四之愈複雜影像，是應該無法藏入更多資料的。其實，本文之方法不是這樣的，圖五為在相同影像品質 34dB 下，四張測試影像在藏入最多資料後的還原圖，以縮小來顯示。圖中 Lena 可藏入 6768 位元，Girl 可藏入 8152 位元，Gold 可藏入 9200 位元，Barbara 可藏入 10480 位元。

由圖五可以發現，Barbara 藏入之資料量，比 Lena 還要多，而 Barbara 所還原之影像品質，卻比 Lena 還要好。其主要原因在於視覺遮罩會修飾還原之空間域影像的品質，而 Lena 被視覺遮罩修飾之後，使 Lena 的每個像素值之間，會產生一些差距。如 Lena 的肩膀是平滑的區域，因像素值差距拉大一點點，就滿明顯的。同樣的原理，Lena 的臉、Lena 的背景、和 Gold 上半部的天空，都有相同的現象。而 Barbara 能量變化較急速，故 Barbara 的每個像素值之間的差距，將會拉的更大，然而卻造成 Barbara 影像有邊緣的地方，都變得很清晰，反而感覺沒有嚴重的破壞。同樣的，Girl 的背景有許多不明顯之邊緣錯綜複雜，假如 Girl 沒有經過視覺遮罩修飾的話，相信 Girl 背景之被破壞的感覺，和 Lena 背景之被破壞的感覺，其實是滿相同的。然而 Girl 背景卻經過視覺遮罩修飾之後，使 Girl 被破壞的程度降低，因此 Girl 背景畫面就會變得比 Lena 之背景還要更清楚，同理可證，Gold 下半部房子，也是如此。

經過上述之描述後，我們可以產生一個論點，並不是資料藏入愈多，其產生的破壞就愈大，而將要衡量此張影像中，能量變化分佈的情況，和視覺遮罩修飾之影響，才可說明此還原影像之被破壞的程度。因此，本文提出「資料藏入」之方法，是較適用於複雜影像的，並且此複雜影像之「最多藏入資料量」，是可以非常多的，此時，本文提出的「資料藏入」之方法，所引起的藏入之破壞對於還原影像而言，也都不怎麼嚴重，同時，我們都可以發現，本文方法之所還原的影像品質都很不錯。

本文方法和 Jpeg-Jsteg 比起來，Jpeg-Jsteg 是一種資料隱藏工具，將資料藏入於 JPEG 之量化後的頻率域係數中，同時，此係數必須不為 0、1、或 -1，則資料才可以藏入於此係數之最不重要位元裡面。然而，此種藏入之方法仍有許多的缺陷，一般而言，在 JPEG 之量化完後的係數幾乎都全為 0，很少有大大於 1 或小於 -1 以上的，所以符合此種條件之係數的個數是很少的，當然，能藏入的資料



量也就必須變的很少，而且此資料量也必須受限於能藏入之係數個數的多寡。基於此，和本文提出「資料藏入」之方法比起來，本文方法是佔有絕對優勢的，因為本文方法有善加利用「層較佳化」之技巧，可以快速地調整  $br$  值大小，使最後一層之  $R_0$ 、 $R_1$ 、 $R_2$  的位元流長度，可以很快地找到等於所要藏入的資料量，並且，此藏入之資料量並沒有受限於某種特定的環境之因素，只要將  $br$  值調整適當的值之後，此藏入之資料量也就等於我們希望所可以達到的資料量，同時，這種調整之方式是不用花費任何計算上的時間。

## 五、 結論

本文提出的「資料藏入」之方法，有善

加利用「層較佳化」之技巧，使可以非常快速的找到我們所要藏入的資料量。而且，經本文之實驗結果顯示，測試影像在還原影像品質 34dB 下，此可以藏入的最多位元數都高於 6700 位元以上，Barbara 將可達到 10000 位元以上，其結果暗示了，若有將藏入資料重新作壓縮，然後再藏入 Barbara 的話，此真正可以藏入的資料量，是比實驗數據更高的，且高於好幾倍以上的，並且，所藏入之資料是藏於 EBCOT 之編碼後的位元流中的，是非常具有隱蔽性的。所以，本文方法比 Wu 與 Tsai[8]提出的「資料藏入」之方法、和 Jpeg-Jsteg[12]、和 JPEG200 之淨水印技術 [3][4][5]應用於「大量資料藏入」之領域方面比起來，本文方法都是比較有優勢的。

表四：在臨界區間裡位元率值、 $br$  值、和最多資料量之間的互動關係

Lena					Girl				
位元率 (bpp)	$br$ 值 (bpp)	資料量 (bytes)	資料量 (bits)	百分比 (%)	位元率 (bpp)	$br$ 值 (bpp)	資料量 (bytes)	資料量 (bits)	百分比 (%)
0.1	0.093	75	600	2.2888	0.12	0.098	93	744	2.3651
0.12	0.07	311	2488	7.9091	0.15	0.08	525	4200	10.681
0.14	0.066	476	3808	10.376	0.18	0.07	823	6584	13.953
0.16	0.06	696	5568	13.275	0.21	0.058	921	7368	13.384
0.18	0.056	779	6232	13.207	0.23	0.055	978	7824	12.977
0.2	0.05	846	6768	12.909	0.25	0.054	1019	8152	12.439

Gold					Barbara				
位元率 (bpp)	$br$ 值 (bpp)	資料量 (bytes)	資料量 (bits)	百分比 (%)	位元率 (bpp)	$br$ 值 (bpp)	資料量 (bytes)	資料量 (bits)	百分比 (%)
0.17	0.12	219	1752	3.9314	0.35	0.25	283	2264	2.4676
0.2	0.085	459	3672	7.0038	0.4	0.19	611	4888	4.6616
0.25	0.07	816	6528	9.9609	0.45	0.15	824	6592	5.5881
0.3	0.06	884	7072	8.9925	0.5	0.13	902	7216	5.5054
0.35	0.051	1001	8008	8.728	0.55	0.1	1025	8200	5.6874
0.4	0.05	1150	9200	8.7738	0.6	0.1	1310	10480	6.663



Lena



Girl



Gold



Barbara

圖五：在相同影像品質 34dB 下藏入最多資料量後所還原的影像

### 參考文獻

- [1] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 Still Image Coding System: An Overview," *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 4, November 2000, pp. 1103-1127.
- [2] T. Ebrahimi, D. Santa Cruz, J. Askelöf, M. Larsson, and C. Christopoulos, "JPEG 2000 Still Image Coding Versus Other Standards," *Proceedings of the SPIE's 45th Annual Meeting, Applications of Digital Image Processing XXIII*, Vol. 4115, 2000, pp. 446-454.
- [3] P. Meerwald and A. Uhl, "Survey of Wavelet-domain Watermarking Algorithms" *Proceedings of the SPIE's 45th Annual Meeting, Security and Watermarking of Multimedia Contents III*, Vol. 4314, 2001, pp.505-516.
- [4] P. C. Su, H. M. Wang, and C. J. Kuo, "Digital Watermarking on EBCOT Compressed Images," *Proceedings of SPIE's 44th Annual Meeting, Applications of Digital Image Processing XXII*, Vol. 3808, October 1999, pp. 313-324.
- [5] P. C. Su, H. J. Wang, and C. J. Kuo, "An Integrated Approach to Image Watermarking and JPEG-2000 Compression," *Journal of VLSI Signal Processing*, Vol. 27, 2001, pp. 35-53.
- [6] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *Proceedings of International Conference on Image Processing*, Vol. 3, 1999, pp. 344-348.
- [7] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Transactions on Image Processing*, Vol. 9, No. 7, July 2000, pp. 1158-1170.
- [8] D. C. Wu and W. H. Tsai, "Spatial-domain Image Hiding Using an Image Differencing," *IEE Proceedings of Vision, Image and Signal Processing*, Vol. 147, No. 1, pp. 29-37.
- [9] ISO/IEC JTC1/SC29/WG1, ISO/IEC FCD 15444-1: Information technology - JPEG2000 image coding system: Core coding system, WG1 N 1646, March 2000, <http://www.jpeg.org/FCD15444-1.htm>.
- [10] ISO/IEC JTC1/SC29/WG1, ISO/IEC FCD 15444-5: Information technology - JPEG2000 image coding system: Reference software, WG1 N 1972, January 2000.
- [11] JJ2000 Project, Software and test data: A Java implementation of the JPEG 2000 encoder/decoder, <http://jj2000.epfl.ch/>.
- [12] D. Upham, Jpeg-Jsteg, <http://www.tiac.net/users/korejwa>.