

一個簡單而有效的 JPEG2000 Codec 浮水印技術

A Simple and Efficient Watermarking Technique Based on JPEG2000 Codec

陳同孝

Tung-Shou Chen

國立台中技術學院 資訊管理系

Department of Information Management,
National Taichung Institute of Technology
404 台中市三民路三段一二九號

No. 129, Sec. 3, San-min Rd., Taichung,
Taiwan 404, R.O.C.

e-mail: tschen@ntit.edu.tw

陳建國

Jian-Guo Chen

國立台中技術學院 資訊管理系

Department of Information Management,
National Taichung Institute of Technology
404 台中市三民路三段一二九號

No. 129, Sec. 3, San-min Rd., Taichung,
Taiwan 404, R.O.C.

e-mail: notebook500@pchome.com.tw

摘要

本篇論文所提出的浮水印技術是於 JPEG2000 Codec 流程中直接藏入與萃取，使藏入與萃取的整個過程更有效率。藏入過程是先以 Torus Automorphisms 技術將浮水印打散之後，再於 JPEG2000 的量化階段之後藏入浮水印，並在藏入時利用失真補償技術，使藏入浮水印後的 JPEG2000 壓縮影像品質失真程度降低。此提出的浮水印技術最大的特點是藏入與萃取的方法簡單易於實作，且於 JPEG 壓縮、模糊、清晰、強調邊緣、馬賽克、均化等的影像攻擊下，仍可偵測出浮水印。

關鍵詞：JPEG2000、Torus Automorphisms、失真補償

Abstract

In this paper, we proposed a watermark scheme that is directly embeds and extracts during JPEG2000 Codec. The embedding algorithm is first to scatter watermark by Torus Automorphisms technology, and then we embed the watermark in the Quantization of JPEG2000 with use Distortion Reduction, so that to reduce the quality distortion degree of embedded JPEG2000 compressed image. The main characteristic of the watermark scheme is simple and easy to implement in this paper, and it can survive on the attacks include JPEG, blurring, edge enhancement, mosaic, averaging, and so on.

Keywords: JPEG2000, Torus Automorphisms, Distortion Reduction

一、前言

隨著科技的進步，數位相機的 CCD 解析程度已從原本的幾十萬像素成長為幾百萬像素，而目前最高階的數位相機 CCD 解析程度以達到四百多萬像素，所以數位相機上的儲存數位影像的記憶體容量也就相對變大，以一張 2048×2048 (約 4 百萬像素) 大小的彩色數位影像為例，其所須的記憶體容量等於 $2048 \times 2048 \times 24 \text{ Bits} = 2048 \times 2048 \times 3 \text{ Bytes} = 12 \text{ Mega Bytes}$ ，如此大記憶體容量，如果以一片數位相機隨機附贈的 8 Mb 記憶卡，根本連一張彩色數位影像都無法儲存，因此影像壓縮技術就變得非常重要。

目前較常見的影像壓縮標準為 JPEG，但因 JPEG 影像壓縮標準為 DCT-Based，其於低壓縮位元率時會使影像產生區塊效應，使影像看起來不協調。而新一代的 JPEG2000 影像壓縮標準[2,3]為 Wavelet-Based，其於低壓縮位元率時不會產生區塊效應，使影像壓縮品質改善許多。因此，本篇論文就採用了 JPEG2000 壓縮標準為浮水印藏入的對象，因其具有較 JPEG 為佳的漸進傳輸以及 JPEG 所沒有的 ROI (Region of Interest) 技術，以致於新一代的 JPEG2000 壓縮標準將可能成為 JPEG 的取代品，因此 JPEG2000 將比 JPEG 更有發展的空間及更高的利用率等，所以於 JPEG2000 壓縮影像上的著作權問題也就更重要了。

目前學者所提出的 JPEG2000 之浮水印技術，藏入的領域大多為頻率域的不可視浮水印[4]，如 P.-C. Su 等人[6]與 P. Meerwald[5]提出的方法，因為頻率域中的頻帶各代表了這張影像的不同特性，所以可針對這些頻帶的重要性來藏入浮水印，讓肉眼不易察覺浮水印的存在，且頻率域比空間域具有較佳的強韌性[4]。

Po-Chyi Su 等人提出的技術，藏入的過

程是於 JPEG2000 的量化階段之後，先以一私鑰 (secret key) 產生一組長度與一個 EBCOT 的編碼區塊 (code block) [2,3,7]之係數個數相同的隨機雜訊序列 (random noise sequence) [4] 來當作要藏入的浮水印，此隨機雜訊序列是由亂數產生器所產生的數列 (由 1 與-1 組成)，之後在每一個編碼區塊中選擇 $\geq$ 門檻值 q (threshold value) 的係數為浮水印藏入的重要係數。而萃取的過程是在熵編碼 (entropy coding) 之後，選擇 $\geq q$ 值之係數為浮水印偵測的重要係數，將這些重要係數利用相關性偵測 (Correlation Detection) 的方法[6]來偵浮水印，因此需要利用原始浮水印序列來與重要係數進行運算，所以此方法並不是非常地理想。

P. Meerwald 提出的技術，藏入的過程也是於 JPEG2000 的量化階段之後藏入，但其是屬於盲的浮水印技術 (blind watermarking) [4]，因此萃取的過程不需要原始浮水印序列。此技術在藏入時，以 QIM (quantization index modulation) 技術[5]來藏浮水印，而萃取的過程是於熵編碼之後，利用 QIM 量化函數[5]來萃取浮水印資料。此技術雖然萃取時的速度較 P.-C. Su 等人提出的技術快，但是其可藏入的資訊量太少。

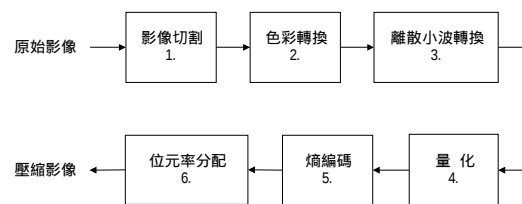
為了避免上述兩位學者的缺點，本篇論文提出一個盲的浮水印技術，其在取出浮水印時不須經過任何複雜的數學運算，因此可以快速且有效率的偵測出浮水印，另外，可藏入的浮水印資訊量也比 P. Meerwald 學者提出的技術多。

本篇論文的組織如下：第一章為前言、第二章為 JPEG2000 Codec 之簡介、第三章為浮水印系統的流程、第四章為實驗結果、第五章為結論。

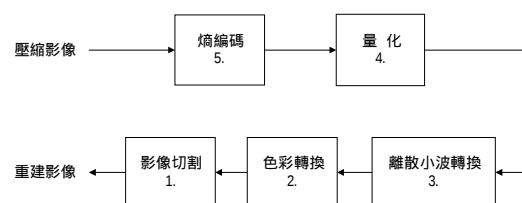
二、JPEG2000 Codec

JPEG2000 是由 ISO (International Standardization Organization) 和 IEC (International Electrotechnical Commission) 此二大標準組織所組成的，其主要是運用於數位影像壓縮標準上。JPEG2000 數位影像壓縮標準規格及流程被詳細定義於 JPEG2000 Part I Final Committee Draft[2]中，其可壓縮彩色、灰階及混合 (含圖形及文字) 的影像，並將無失真 (Lossless) 及有失真 (Lossy) 壓縮整合為一，其有失真壓縮之壓縮比及影像之重建品質皆比 JPEG 數位影像壓縮標準為佳。本論文主要是針對有失真壓縮為藏入的環境，因失真壓縮 (具有較高的壓縮比) 於網路上的利用率較高。JPEG2000 壓縮標準主要包含了六個程序 [2,3]：1. 影像切割 (Tiling)、2. 色彩轉換 (Component Transform)、3. 離散小波轉換

(Discrete Wavelet Transform)、4. 量化 (Quantization)、5. 熵編碼 (Entropy Coding)、6. 位元率分配 (Rate Allocation)。圖一及二分別為 JPEG2000 的壓縮及解壓縮流程圖。



圖一 JPEG2000 壓縮 (Encoder) 流程圖



圖二 JPEG2000 解壓縮 (Decoder) 流程圖

三、系統架構與流程

本篇論文中的浮水印技術，是於 JPEG2000 壓縮 (Encoder) 流程中的量化階段與熵編碼階段之間進行藏入浮水印資料，因於量化階段之後的係數皆已正規化為固定的 32 位元精確度[2]，所以實作上較為容易。本篇論文提出的浮水印技術，分成浮水印藏入系統與浮水印萃取系統這兩部分來探討。

(一) 浮水印藏入系統

首先將要藏入的浮水印圖案以 Torus Automorphisms 技術[1]打散，使藏入的浮水印更具保密性。(1)為 Voyatzis 與 Pitas 兩位學者所提出的單一參數 Torus Automorphisms 技術

[8]，其中的 $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ 為浮水印圖案的原始座標，

而 $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ 為打散後的座標， N 為浮水印圖案的長

寬， n 為轉換矩陣 $\begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}$ 與 $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ 的疊代次數，其 n 值我們定義為打散私鑰。

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^n \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \bmod N \quad (1)$$

將長寬為 32×32 的浮水印圖案以下列的

Torus Automorphisms運算式打散：

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}^n \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \bmod 32$$

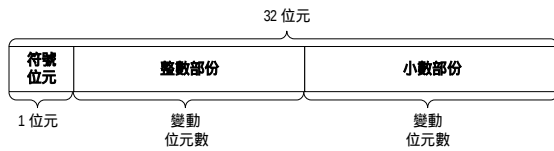
當打散私鑰 n 等於3、8、16及24時的圖案變化，如圖三所示，從圖中可發現當打散私鑰 $n=24$ 時會出現與原圖一樣的情況，此即為週期數 T ，所以用於還原浮水印的疊代次數就等於 T 減打散私鑰 n ，在此我們定義為還原私鑰。



圖三 打散後的浮水印變化

打散後的浮水印藏於量化階段之後的頻帶中，而要藏入的頻帶數，需視浮水印的大小(size)而定，依 LL_j 、 LH_j 、 HL_j 、 HH_j 與 $j = J \dots 1$ (J 為DWT的分解次數)的順序來藏入，以一張 512×512 的影像，在經過5次的DWT分解後，其 LH_j 、 HL_j 與 HH_j 頻帶為 $\frac{512}{2^j} \times \frac{512}{2^j}$ 的大小 ($j = 1 \dots 5$)，而 LL 頻帶為 $\frac{512}{2^5} \times \frac{512}{2^5} = 16 \times 16$ ，所以當藏入的浮水印大小為 32×32 時，則須 LL 、 LH_5 、 HL_5 與 HH_5 四個頻帶才可完全藏入。

在JPEG2000的量化階段之後，每個頻帶的係數皆正規化成圖四中的形式，而我們將浮水印元素藏於係數中的整數部份，因小數的部份相對整數部份而言是較不是重要的。然後直接修改整數部份中某位置的位元，我們定義為 M_b ，當 M_b 的位元值改為1則表示藏入1值，反之則藏0值，而此 M_b 由公式(2)求得，(2)中的整數位元數就是圖四中整數部份的變動位元數， α_b 為使用者自定的門檻值，值愈大則藏入的浮水印愈強韌，反之則愈脆弱。



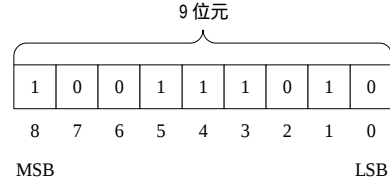
圖四 正規化後的係數形式

$$M_b = \lfloor \text{整數位元數} \times \alpha_b \rfloor \quad (2)$$

$$\alpha_b \in [0, 1)$$

浮水印資料藏於較重要的頻帶 (如 LL 頻帶)，當然相對的會影響影像品質的失真程度，因此我們就採用失真補償技術[1]，來減低品質的失真度。以下我們舉例說明失真補償技術：

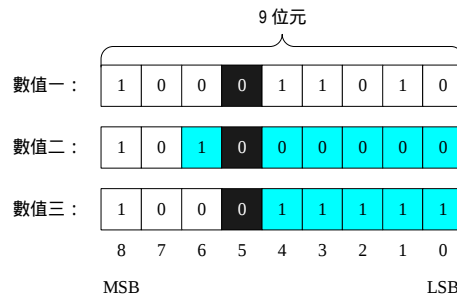
假設原始係數值為 314_{10} ，其二進制格式如圖五所示，我們將圖五中的最不重要位元 (Least Significant Bit, LSB) 定義為第0位元，而最重要位元 (Most Significant Bit, MSB) 定義為第8位元，而我們要藏入的浮水印資料為二元值，其值定義為0與1。為了使藏入浮水印後的係數值接近於原始係數值，我們將藏入0與1時的情況分成兩部分來探討：A.藏入0時的情況、B.藏入1時的情況。



圖五 314_{10} 之二進制表示圖

A. 藏入0時的情況

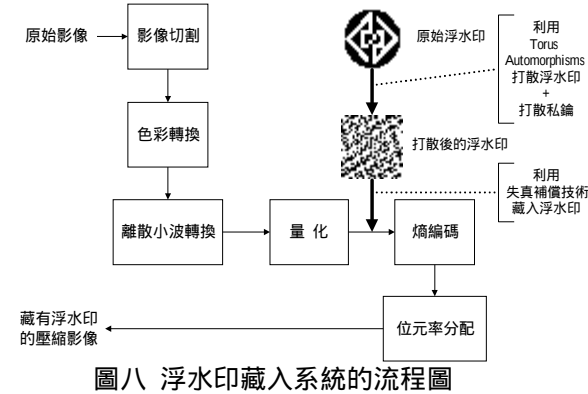
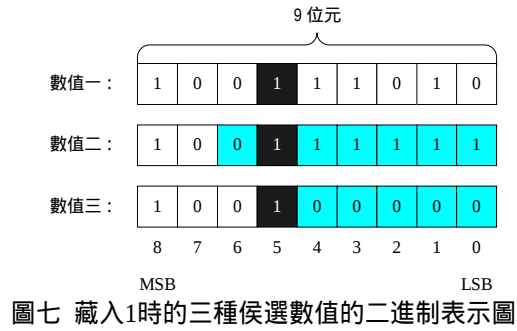
假設我們於原始係數的第5位元藏入0，會使修改後的數值與原始係數值相差 $2^5 = 32$ ，如圖六中的數值一，其值為 282_{10} ，而為了降低失真程度，我們選擇一個在第5位元為0且其值最接近原始係數值的數值來代替原始係數。但可選擇的數值太多，因此為了簡化選擇的過程，我們只考慮三種數值，第一種為數值一，第二種是把數值一的第6位元改為1，而第4、3、2、1、0位元改0，如圖六中的數值二，其值為 320_{10} ，而第三種是把數值一的第4、3、2、1、0位元改為1，如圖六中的數值三，其值為 319_{10} ，最後我們只須從此三種數值中選擇與原始係數值最接近的數值來替代原始係數即可，因此就能降低影像的品質失真度。



圖六 藏入0時的三種候選數值的二進制表示圖

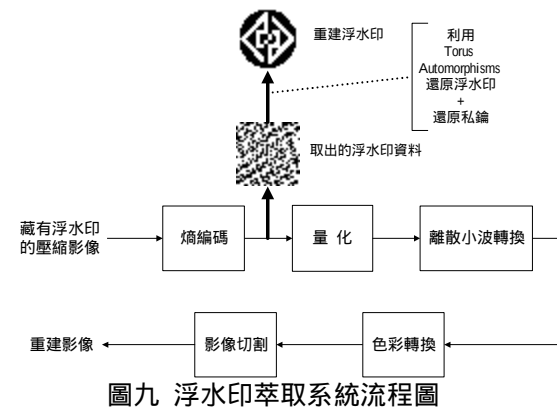
B. 藏入1時的情況

此部份的修改方式與A.部份的修改規則一樣，但所須修改的位元值與A.部份相反，修改後的三種候選數值如圖七所示。圖八為整個浮水印藏入系統的流程圖。



(二) 浮水印萃取系統

浮水印的萃取是於JPEG2000解壓縮(Decoder) 流程中的熵編碼之後進行取出，只須根據(2)中之 M_b ，再依藏入浮水印時的頻帶順序，萃取至係數總數等於原始浮水印的大小為止，之後再利用Torus Automorphisms技術與私鑰來還原取出的浮水印資料。圖九為浮水印萃取系統的流程圖。



四、實驗結果

本系統的實驗平台是 INTEL P -500 CPU、192MB RAM 的筆記型電腦，作業系統是 Microsoft Windows 2000 Professional，撰寫浮水印系統程式的發展工具為 Java Builder 5.0，而使用的 JPEG2000 Codec 是採用 JJ2000 提供的 4.1 版原始程式碼[9]。

在實驗結果分析，是利用 PSNR (Peak Signal to Noise Rate) 尖峰訊雜比[1]，來當作評估重建影像與原始影像之相似度的指標，PSNR 的單位為 dB，當所算出的 PSNR 之 dB 愈大，表示原始影像與重建影像相似度愈高，而在萃取出的浮水印品質評估，是採用 BCR (Bit Correct Ratio) 位元正確率 [1]，其是以百分比為單位，當百分比愈高表示取出的浮水印愈清楚，反之則較不清楚。本篇論文之實驗，在 (1)中的打散私鑰 n 為使用者設定的 key 值，我們以 8 為實驗值，而為了簡化(1)中的轉換矩陣，我們設 k 值為 1。在(2) 中之 α_b 於 LL 頻帶採用 0.4 LH 及 HL 頻帶採用 0.4 HH 頻帶採用 0.35， α_b 設得太大，雖然可抵抗嚴重的壓縮破壞，但對整張影像的品質破壞就愈嚴重。

P. Meerwald 提出的浮水印技術，使用三張 512×512 的灰階影像，分別為 “Lena”、“Goldhill”、“Fishing Boat”，並以 JJ2000 為 JPEG2000 Codec，在其內定值[9]的環境下，以不同的視窗與子區塊大小[5]來藏入浮水印。這三張影像在不同的視窗與子區塊大小下所能藏入的位元數與 PSNR，如表一所示。

我們提出的浮水印技術，以三張 512×512 的灰階影像，分別為 “Lena”、“Goldhill”、“Fishing Boat”，如圖十的 (a)、(b)、(c) 所示，及 32×32 大小的 “NTIT”黑白浮水印，如圖十的(d)。我們利用 JJ2000 的內定值為環境，藏入浮水印於這三張影像中，其實驗結果如表二所示。從表二中可以發現我們提出的方法比 P. Meerwald 提出的技術的藏入位元數與 PSNR 值高，所以可看出我們的方法，所引起的藏入破壞是很小的。

表一 P. Meerwald 提出的方法之實驗結果

影 像	視窗大小	子區塊大小	藏入位元數	PSNR
Lena	4	64	85	32.05 dB
Fishing Boat	2	32	194	31.45 dB
Goldhill	2	16	383	32.09 dB

表二 我們提出的方法之實驗結果

影 像	藏入位元數	PSNR
Lena	1024	42.18 dB
Fishing Boat	1024	41.49 dB
Goldhill	1024	41.67 dB



圖十 (a) “Lena” (512×512) 原始影像、(b) “Fishing Boat” (512×512) 原始影像、(c) “Goldhill” (512×512) 原始影像、(d) “NTIT 圖騰” (32×32) 黑白浮水印

接下來我們將圖十之(d)藏入於圖十之(a) (b) (c)之後，我們對這些影像做 JPEG2000 之 0.25 bpp、0.1 bpp 壓縮攻擊，其實驗結果如圖十一所示。我們從圖十一中，可發現壓縮位元率在 0.25 bpp 的壓縮攻擊下，其浮水印仍可完全正確地取回來，如圖十一之(b)、(f)、(j)所示，而當壓縮攻擊低於 0.25 bpp 時，則其取出的浮水印，如圖十一的(d)、(h)、(l)所示，我們發現這些浮水印開始出現少許的位元錯誤，但其浮水印之 BCR 皆大於 85%，這也就是代表只有少許的位元錯誤，而大部份的位元都是正確的，所以我們的浮水印技術是可抵抗嚴重的 JPEG2000 之壓縮攻擊，而這種程度的壓縮攻擊已經算是相當嚴重的破壞攻擊。

如上所述，(d)、(h)、(l)之位元錯誤的原因，是因為受到 JPEG2000 之 EBCOT[2,3,7] 的影響。在 EBCOT 中，是採用位元平面 (bit plane) 之編碼方式，也就是針對每一個位元平面來作編碼，所以最先編碼之最高位元平面，其編碼出來的位元流為最重要的位元流，然後 JPEG2000 為了完成壓縮位元率，而必須丟棄 (truncation) 大量較不重要的位元流，也就是說會丟棄較低位元平面所產生的位元流。所以我們所藏入的位元平面所產生的位元流，有可能會被 EBCOT 視為不重要位元流，故這些所藏入的位元就會遭 EBCOT 的丟棄，而導致位元之錯誤。

最後我們在 1bpp 的 512×512 “Lena”灰階影像中藏入浮水印，然後對這張影像做許多不同的影像攻擊，來探討我們的方法是否具有強韌性，而這些影像攻擊包含：模糊 (Blur)、清

晰 (Sharpen)、強調邊緣 (Edge Enhancement)、馬賽克 (Mosaic)、均化 (Averaging) 及 8% 均勻雜點 (Uniform Noise)、切割 (Cropping) 四分之一與 JPEG 壓縮 (壓縮比 20:1) 等攻擊。其攻擊後的影像與萃取出的浮水印如圖十二與十三中所示，我們發現這些取出的浮水印，其 BCR 全部都大於 69%，由此可知我們的方法是具有非常具有強韌性的。

五、結論

本篇論文是建構於新一代的 JPEG2000 靜態影像壓縮標準上的浮水印系統，我們將藏入的浮水印利用失真補償技術，提升藏入浮水印後的影像品質，並將以 Torus Automorphisms 技術將浮水印打散，使藏入的浮水印具有較高的安全性。本系統所採用的技術非常簡單，且所藏入浮水印的影像品質都非常好，並且在許多的影像攻擊下，如 JPEG2000 的壓縮 (0.1bpp，即 1:80 的壓縮比) 模糊、馬賽克、切割、JPEG 壓縮等，我們的方法都可以強韌於這些攻擊。未來我們將嘗試使此浮水印技術更強韌，同時又可抵抗幾何影像攻擊。

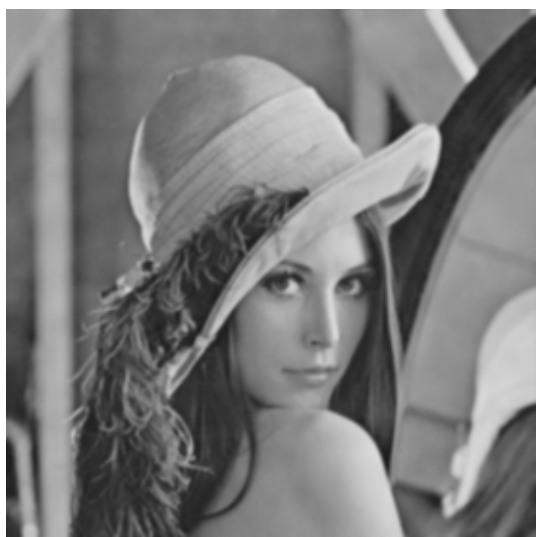
參考文獻

- [1] 張真誠、黃國峰、陳同孝，電子影像技術，松崗電腦圖書資料股份有限公司，2001 年 2 月。
- [2] M. Boliek, C. Christopoulos, and E. Majani, “JPEG2000 Final Committee Draft SC29WG1 N1646,” ISO/IEC JTC1/SC29/WG1 N1646R, April 2000.
- [3] C. Christopoulos, A. Skodras, and T. Ebrahimi, “The JPEG2000 still image coding system: an overview,” *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 4, pp. 1103-1127, November 2000.
- [4] S.-J. Lee and S.-H. Jung, “A survey of watermarking techniques applied to multimedia,” *Proceedings of IEEE International Symposium on Industrial Electronics*, Vol. 1, pp. 272-277, 2001.
- [5] P. Meerwald, “Quantization watermarking in the JPEG2000 coding pipeline,” *Communications and Multimedia Security Issues of The New Century, IFIP TC6/TC11 Fifth Joint Working Conference on Communications and Multimedia Security*, pp. 69-79, May 2001.
- [6] P.-C. Su, H.-J. Wang, and C.-C. Jay Kuo, “An Integrated Approach to Image Watermarking and JPEG-2000 Compression,” *Journal of VLSI Signal Processing*, Vol. 27, pp. 35-53, 2001.

- [7] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Transactions on Image Processing*, Vol. 9, No. 7, pp. 1158-1170, July 2000.
- [8] G. Voyatzis and I. Pitas, "Chaotic Mixing of Digital Images and Applications to Watermarking," *Proceedings of European Conference on Multimedia Applications, Services and Techniques*, Vol. 2, pp. 687-695, May 1996.
- [9] JJ2000, "JJ2000 An introduction to the software," <http://jpeg2000.epfl.ch>.



圖十一 “Lena” (a)壓縮位元率 0.25 bpp (PSNR 34.75 dB)、(b)在(a)中取出的浮水印 (BCR 100%)、(c)壓縮位元率 0.1 bpp (PSNR 30.27 dB)、(d)在(c)中取出的浮水印 (BCR 93.07%)， “Fishing Boat” (e)壓縮位元率 0.25 bpp (PSNR 31.47 dB)、(f)在(e)中取出的浮水印 (BCR 100%)、(g)壓縮位元率 0.1 bpp (PSNR 27.74 dB)、(h)在(g)中取出的浮水印 (BCR 86.72%)， “Goldhill” (i)壓縮位元率 0.25 bpp (PSNR 31.63 dB)、(j)在(i)中取出的浮水印 (BCR 100%)、(k)壓縮位元率 0.1 bpp (PSNR 28.69 dB)、(l)在(k)中取出的浮水印 (BCR 93.55%)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

圖十二 (a)模糊後的影像、(b)清晰後的影像、(c)在(a)中取出的浮水印 (BCR 84.86%)、(d)在(b)中取出的浮水印 (BCR 92.68%)、(e)強調邊緣的影像、(f)馬賽克的影像、(g)在(e)中取出的浮水印 (BCR 69.73%)、(h)在(f)中取出的浮水印 (BCR 73.63%)



圖十三 (a)均化後的影像、(b)加入 8%均勻雜點後的影像、(c)在(a)中取出的浮水印 (BCR 82.42%)、(d)在(b)中取出的浮水印 (BCR 70.11%)、(e)切割四分之一的影像、(f) JPEG 壓縮 (壓縮比 20 : 1) 的影像、(g)在(e)中取出的浮水印 (BCR 79.39%)、(h)在(f)中取出的浮水印 (BCR 87.79%)