

有關碎形壓縮影像之數位浮水印方法

Digital Watermarking on Fractal Compressed Images

杜維昌
義守大學資訊工程系
wcdu@isu.edu.tw

陳俊廷
義守大學資訊工程系
m9503035@stmail.isu.edu.tw

摘要

近年來，數位多媒體技術隨著電腦與網路迅速發展，同時漸漸取代傳統紙本的傳播媒介。當每個人可任意接收或傳播媒體時，如何保護智慧財產權自然成為迫切解決的問題，而數位浮水印技術是有效保護媒體的方法之一。一般會將原創者具代表性的圖騰或商標作為浮水印，利用嵌入技術隱藏在數位媒體中，以證明其合法性。將來若有所權上的爭議，可萃取浮水印作為版權歸屬的重要依據。此外，在眾多數位影像儲存格式中，碎形壓縮是一種相當特殊的儲存方式，具有解析度獨立、高壓縮比、高影像品質等優點。本文即以碎形壓縮影像為標的，設計並實作相關數位浮水印方法，分析比較不同嵌入方法對於影像品質的差異。

關鍵詞：碎形影像壓縮、數位浮水印、智慧財產權保護。

1. 簡介

在這知識爆炸的年代，大量的資訊伴隨著多媒體的發展，使得以往的傳播媒介已不敷應付。也因為多媒體數量的日漸增加，相關的正負影響面逐漸被各界重視，例如保護機制的問題或是非法儲存、散佈等議題，而數位浮水印(digital watermarking)技術便是有效解決智慧財產權保護問題的方法之一。數位浮水印技術於1990年首次被提出，是在多媒體資料中嵌入版權相關的浮水印資訊，以宣示媒體所有權或認證的主要依據[2]。和一般加密技術不同的是，數位浮水印技術通常會讓人類感官難以察覺到其中的微小變化，通常需具備不易察覺(imperceptibility)、不易移除(undeletable)、不易計量察覺(statistically undetectable)、強韌性(robustness)、明確性(unambiguous)等特性[3]。

以資訊理論的角度，一般數位浮水印嵌入方法主要是針對原始媒體作細微的修改，為了保有原影像之品質而選擇不重要的部份或降低修改的幅度。但修正的幅度過小，欲取出之

浮水印容易被干擾而影響萃取的成效。因此，如何在影像品質與抵抗攻擊之間作取捨是數位浮水印技術需考量的議題。從影像嵌入數位浮水印的場所來看，可概分成兩大類：第一類是空間域方法，亦即直接修改原始影像的像素明亮值來達成嵌入浮水印之目的。利用這類方法可具有快速運算的優點，但相較於頻率域方法往往無法有效抵抗各類攻擊。相關文獻如LSB (least significant bits)演算法[4]、迴旋擾亂(toral automorphism)[5]即可把浮水印經有限次的轉換後成為雜訊，但再經一定的轉換週期，又可還原成原始數位浮水印。第二類是頻率域方法，亦即將數位資料轉換成頻率域後，藉由改變頻率域中的係數來嵌入浮水印，再轉回空間域。此種方法具有較佳的抵抗攻擊能力，但也需要較大的運算時間。例如應用離散餘弦轉換或是離散小波轉換[21]皆是屬於頻率域的範疇。

多媒體靠著網路迅速傳播至各地，然而這些資料量往往很大，若不以壓縮形式作傳輸將對有限頻寬產生很大的負擔，因此要如何減少多餘的資料量並仍能維持原媒體品質自然成為重要議題。在影像壓縮技術方面，碎形壓縮是利用影像本身自我相似的原理，經由收斂仿射轉換(contractive affine transformation)得到碎形碼，以達壓縮之效果。碎形影像壓縮具有解析度獨立(resolution independence)、高壓縮比(high compression ratio)、高品質(highly reconstructed image quality)等優點，但在傳統編碼上進行全域搜尋(full search)比對需耗費大量計算時間而影響其應用層面。因此，有許多學者針對碎形壓縮提出一些改良的方法[19][20][21]，無論在碎形編碼速率或影像品質提升都有不錯的成效。

碎形影像壓縮的應用層面很廣，可運用於特徵分類、去除雜訊、視訊編碼等領域[15][16]。經實驗測試。其壓縮比甚至可達10000:1以上[1]，因此若在碎形壓縮的基礎下，設計適切的數位浮水印技術自然可以有效

保護其版權。在嵌入過程中，影像品質難免會有所犧牲，但仍希望加入浮水印的同時仍保有較小的失真。目前相關的研究皆以碎形碼作為浮水印嵌入的場所，有些學者提出在搜尋最佳定義域區塊的過程中，依據浮水印資訊限制搜尋範圍，進而搜尋出局部最佳的碎形碼，例如嵌入於位置參數[10][18][17]、旋轉參數[11]或是灰階參數上作嵌入[13]，此類作法的優點是可以節省搜尋的時間，同時其失真不致於太多。另外，有些學者直接修改碎形碼的部分參數[12][14]，此種作法會降低較多的影像品質。在此一研究中，主要探討如何在碎形壓縮的過程中嵌入數位浮水印，我們將採局部搜尋策略，分析是否隨著嵌入位置區塊的不同對於影像品質的影響程度，對此將以真實影像作測試來展現實際的成效。

2. 碎形影像壓縮

碎形的概念雖然早已被提出，但直到 Benoit Mandelbrot 在 1975 年再次提出碎形理論才引起科學界的注意。1989 年，Barsley 首先將碎形理論運用在影像壓縮上[6]，然而此一演算法並沒公諸於世，直到 1992 年由博士班學生 Jacquin 公開其碎形影像壓縮技術[7][8]，爾後才有較廣泛地發展，包括與小波轉換或向量量化相結合。碎形影像壓縮是屬可失真的靜態影像壓縮，其技術是利用影像本身自我相似的原理，經由收斂仿射轉換得到碎形碼，以達壓縮之效果。此一技術是建立在自然影像具有局部自我相似之特性。此一精神與向量量化編碼法(vector quantization, VQ)很類似，都是建立在區塊編碼(block encoding)的架構上，也就是針對每一個已分割之區塊加以處理。而兩者主要差別在於 VQ 編碼是用固定的編碼簿(codebook)，碎形壓縮的參考來源則是影像本身，雖然不需用到編碼簿，但在比對搜尋最佳匹配區塊時需要較多的計算時間，這編

碼方法統稱為局部疊代函數系統(partitioned iterated function system)。

2.1 碎形影像編碼

一般自然影像經常具有局部自我相似特性，圖一中可發現影像中雖然大小區塊不同但紋理卻很相似。在碎形影像壓縮的過程中，首先將原影像分成兩種區塊：值域區塊(Range Block，簡稱 RB)和定義域區塊(Domain Block，簡稱 DB)。值域區塊為 $B \times B$ 大小的不重疊方塊，定義域區塊通常設為值域區塊的四倍大小且區塊可部分重疊。假設 256×256 大小的影像，以 8×8 之 RB 對原影像作切割，共可得到 $(256/8) \times (256/8) = 1024$ 個不重疊的 RB，而這些 RB 將作為碎形編碼的單位，RB 所構成的集合稱為值域池(Range Pool)。另一方面，對原影像以大小 16×16 作分割，則可得 $(256-16+1) \times (256-16+1) = 58081$ 個可部分重疊的 DB，其集合稱為定義域池(Domain Pool)。然後對每個值域區塊 R_i 而言，可在 Domain Pool 中找到一個最佳 DB 和一疊代函數 W_i ，使得 $W_i(D_j)$ 與 R_i 最相似，如圖二所示。

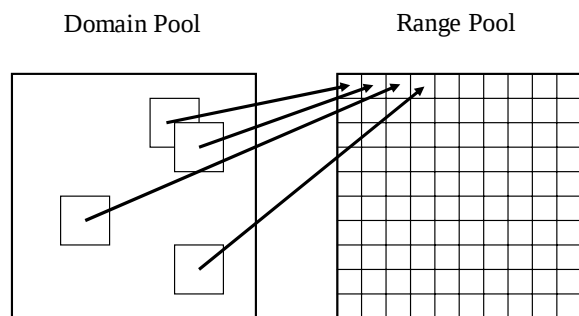
在碎形影像壓縮的編碼中，主要是找到一個 DB 經仿射轉換後能與 RB 最相近。若以三維的座標來描述， x 軸和 y 軸表示影像上每個像素的位置， z 軸用來表示該像素點的灰階值，那麼此仿射轉換可表為

$$t_i \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}$$

其中 e_i 與 f_i 表 DB 在影像中的位置， s_i 為對比度， o_i 為明亮度。係數 a_i 、 b_i 、 c_i 與 d_i 為八個方向的旋轉矩陣，簡化表為旋轉參數 i_i 。因此，一張影像的每個 RB，就可以用 $[e_i, f_i, i_i, s_i, o_i]$ 五項參數來作為儲存的參數，稱之為碎形碼。詳細說明如下：



圖一、灰階影像中的局部相似性



圖二、區塊疊代示意圖

1. 收縮轉換(Shrink Transformation): 將不一樣大小的 DB 和 RB 作比較, 就必須將 DB 中對應的四個像素點平均為 RB 的一個點。
2. 平移轉換(Translation Transformation): 當 DB 經收縮轉換後, 會記錄此區塊之位置。
3. 旋轉轉換(Rotation Transformation): 為了 DB 經轉換後與 RB 能有最佳配對, 同時要減少搜尋時間, 採取下列八種旋轉角度:

$$T_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, T_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix},$$

$$T_3 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, T_4 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, T_5 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$T_6 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, T_7 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

4. 灰階對應轉換(gray-level mapping Transformation): 可透過平均值或最小平方誤差逼近法得到灰階對應參數 s_i 和 o_i 。假設經由前面系列轉換後的某一 DB 擁有最近似的值: $Mean(R_i) = s_i \times Mean(D_j) + o_i$, 只要 $\|s_i\| < 1$, 就會滿足收縮性。

在比對區塊時, 用均方根(MSE)來評估該 DB 是否為 R_i 之最佳配對區塊。當 MSE 越小代表 DB 和 RB 差異度越小。最後, 將每個 RB 所找到最佳 DB 的仿射轉換參數 $[e_i, f_i, i_i, s_i, o_i]$ 記錄起來。此時, 碎形碼即代表全影像之資訊, 包括 DB 的位置、旋轉、對比度、亮度。

2.2 碎形影像解碼

在解壓縮部份, 欲重新建立影像只需將任意一張與原始影像同大小的影像, 用碎形碼以局部疊代函數系統進行數次的疊代, 即每一個 RB 根據碎形碼取出位於 (e_i, f_i) 的 DB, 針對該 DB 作收縮、旋轉、對比和亮度的調整。當全部的 RB 都進行完解碼的過程後, 便算完成一次的疊代。根據使用者設定的疊代次數, 即可得到一張重建後的影像。相較於壓縮過程的耗時, 碎形影像解壓縮過程顯得快速許多。

3. 浮水印嵌入方法

在 1996 年開始有學者先後探討碎形壓縮影像中嵌入數位浮水印。藉由修改部分參數, 在不影響原影像品質的前提下作調整, 以達到嵌入數位浮水印的目的。由於碎形碼允許可以調整的範圍有限, 不易直接對已搜尋的最佳碎形碼作變動, 因此把浮水印嵌入的演算法和碎形壓縮演算法作結合是個不錯的選擇。也因為在碎形壓縮過程中尋找最契合之 DB 的同時把

浮水印嵌入其中, 可以讓人類視覺無法察覺其差異。

在傳統編碼過程, 每個 RB 都要從 Domain pool 中做全搜尋(full search), 從中找到最契合的 DB 來作碎形碼的儲存。但在碎形碼中的位置參數作浮水印的嵌入是利用區域搜尋(local search region, LSR)的概念, 不僅減少搜尋時間, 對於失真度也有限。然而隨著嵌入位置的不同其浮水印之透明度會有所不同, 本研究將對此作分析比較。

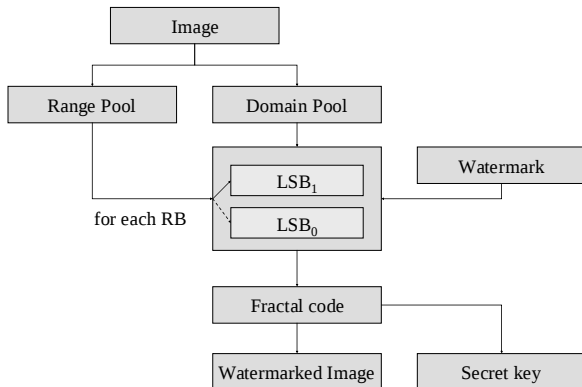
以下先介紹浮水印嵌入流程, 位置參數加入浮水印資訊的相互關係及嵌入流程如圖三所示。在碎形編碼時順便將浮水印的嵌入考量進去, 同時也因為將可參考的區域分成兩部份做嵌入, 與搜尋全部的 DB 相較速度會較快。以下便是嵌入流程:

- 將浮水印打亂, 使之看似無意義的資訊如亂碼或雜訊般。
- 在原影像方面, 如傳統的碎形壓縮一樣, 先將原影像分成兩種區塊: 值域區塊和定義域區塊。
- 將 DB 分成兩部份, 分別為 LSB_0 、 LSB_1 。
- 每一 RB 都各自擁有專屬的碎形碼, 若該 RB 所要嵌入的浮水印為 0, 就只能在 LSB_0 的定義域區塊內去搜尋最佳的 DB; 若 RB 要嵌入的浮水印為 1 時, 在 LSB_1 的區塊內找尋最匹配的 DB。
- 完成碎形壓縮的流程後, 將灰階映對轉換的兩個參數對比度 s_i 和明亮度 o_i 、旋轉參數 i_i 保留成密鑰。
- 把所得到的碎形碼解壓縮回來, 即得蘊含數位浮水印的影像。

4. 浮水印萃取方法

在浮水印取出過程中, 可以利用先前在嵌入浮水印後得到的密鑰 $[i_i, s_i, o_i]$ 和已含浮水印的影像去取得浮水印。簡單來說, 即把密鑰裡的三個參數 $[i_i, s_i, o_i]$ 當作已確定的參數, 位置參數為未知數, 而將密鑰代入搜尋演算法取出最適合的位置參數 $[e_i, f_i]$, 進而取出浮水印判斷是否為擁有者或創作人。圖四為概要流程圖, 萃取浮水印步驟如下:

- 將已含有浮水印之影像同樣分成兩種區塊 DB 與 RB。
- 把密鑰 $[i_i, s_i, o_i]$ 代入搜尋演算法去尋找最適合的位置參數並判斷該 DB 編號的奇偶, 若編號為偶數則判斷為 0, 奇數為 1。



圖三、浮水印嵌入流程圖

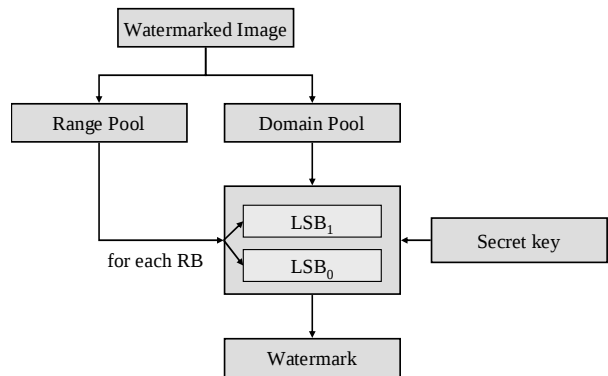
■ 當全部的 RB 都完成最佳搜尋時，含浮水印影像的隱藏資訊即可取得，最後和與原始浮水印作比較，看是否為原創作者或合法擁有者。

5. 實驗結果

首先，以 256×256 大小的 Lena 作測試影像，一般判斷影像品質的優劣或與原始影像的差異是使用訊號雜訊比 PSNR 值，亦即 $PSNR = 10 \cdot \log_{10}(255^2 / MSE)$ 。數值越高則代表品質越佳，也越不容易被發現與原影像有所差異，疊代過程測試結果如圖五。

在碎形壓縮浮水印中，採用 Lena 灰階影像作測試。圖六主要簡單測試在搜尋 DB 時只作一半的搜尋，以觀察影像 PSNR 值是否在可接受的範圍內。圖(a)和(b)分別只有在左半邊與上半部作搜尋，(c)則是在八種旋轉角度中，只提供四種角度作匹配選擇。從 PSNR 值來看，雖然減少可自我參考的區域，但是失真度皆在 32dB 以上，而且編碼速度約為原全搜尋時間的一半。

接著，進一步對位置參數作測試，在設定上以 Lena、Baboon 以及 Cameraman 三張作測試影像其單純作碎形壓縮之 PSNR 分別為 34.56、27.57、29.29dB，並嵌入隨機產生的 0 或 1 代表浮水印，看是否隨嵌入區塊位置的不同對於影像品質有所差異，結果如圖七、圖八、圖九。黃色區塊設為 0，另一區塊就設為 1，也就是說若 RB 要嵌入 0，只能在黃色區塊作搜尋，要嵌入 1 則反之。最初可看到(a)(b)是直接水平和垂直軸對切，可參考的區域僅在上半部或僅在下半部，此方法可能出現的問題就是若 RB 欲嵌入 1，但其最佳參考區塊卻在 0 的參考區域，就有可能因為分割的不平均而降低遷入後的影像品質。因此，分成更密集如



圖四、萃取浮水印流程圖

(c)(d)，其區塊相隔四個像素，由於可參考區塊散佈的較平均，所以其 PSNR 相較於(a)(b)就更為提升。最後的(e)(f)則觀察改成格子狀是否有所不同，在(e)這張是測試大區塊的劃分，雖然影像品質比(c)(d)還要低，但仍比只分成左右區塊還要高，而(f)為最小區塊的格子狀，而圖七的(f)其 PSNR 值很近似於作全區域搜尋的 34.56，顯見其分割最為均勻。

6. 結論

碎形壓縮是一種高壓縮比的影像儲存方式，在本研究中主要去探討利用不同參考區域去對位置參數作嵌入，以比較浮水印在透明度上的差異。從實驗結果可知，分割的兩區域交錯，紋理分佈就越平均，那麼嵌入後影像品質也越不受影響。未來工作將可進一步納入人類視覺的感官因素，依據區塊的視覺敏感動態決定嵌入量的多寡，以期可將嵌入後失真降到最小。

參考文獻

- [1] M. F. Barnsley, "Fractals Everywhere," Academic Press, San Diego, 1988.
- [2] Kutter M. and Petitcolas, F. A. P., "A Fair Benchmark for Image Watermarking System," *Proceedings of SPIE*, vol. 3657, pp. 219-239, Jan., 1999.
- [3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.
- [4] R. G. van Schyndel, A. Z. Tirkel, C. F. Osborne, "A Digital Watermarking," in *Proc. ICIP94, IEE Int. Conf. Image Processing*, vol. 2 pp. 86-90, 1994.

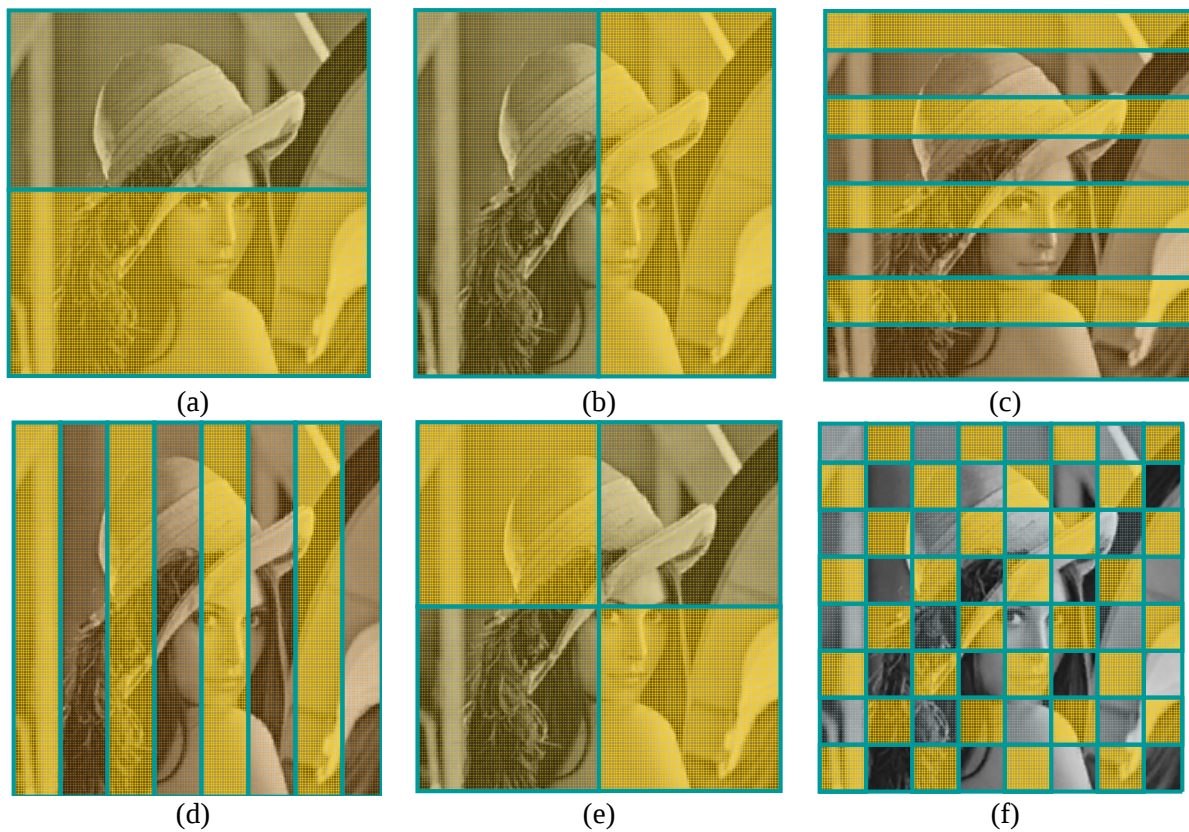
- [5] G. Voyatzis and I. Pitas, "Applications of toral automorphisms in image watermarking," in *Proceedings of the IEEE International Conference on image processing*, vol. 2, pp. 237-240, 1996.
- [6] M. F. Barnsley and A. D. Sloan, "A better way to compress images," *BYTE magazine*, pp. 215-233, Jan. 1988.
- [7] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. on image processing*, vol. 1, no. 1, pp. 18-30, Jan. 1992.
- [8] A. E. Jacquin, "Fractal image coding: A review," *Proc. IEEE*, vol. 81, pp. 1451-1465, Oct. 1993.
- [9] Yao Zhen, "Fixed Point in Fractal Image Compression as Watermarking," *ICIP*, vol. 2, Sept. 2003.
- [10] J. Puatè and F. Jordan, "Using fractal compression scheme to embed a digital signature into an image," in *Proc. SPIE Photonics East Symp.* Boston, MA, Nov. 1996.
- [11] Cheng-Hao Li and Shuenn-Shyang Wang, "Digital watermarking using fractal image coding," *Letter in IEICE Trans. Fundamentals*, vol. E83-A, no.6, pp. 1286-1287, June 2000.
- [12] Wei-Rong Tseng, "Digital Image Watermarking Based on Fractal Compression Scheme," 國立高雄第一科技大學電腦與通訊工程系碩士論文, 民國九十年五月。
- [13] 楊朝源, "A Study of Embedding Digital Watermarks on a Fractal Compressed Image," 育達商業技術學院資訊管理系碩士論文, 民國九十四年六月。
- [14] Ming Hong Pi, Chun Hung Li and Hua Li, "A Novel Fractal Image Watermarking," *Proc. IEEE Trans. on Multimedia*, vol. 8, no. 3, June 2006.
- [15] Ming Hong Pi, Chun Hung Li and Hua Li, "Fractal-Wavelet Image Denoising Revisited," *IEEE Transactions on Image Processing*, vol. 15, no. 9, Sep. 2006.
- [16] M. Ghazel, R. K. Ward, R. Abugharbieh, E. R. Vrscay, and G. H. Freeman, "Simultaneous fractal image denoising and interpolation," *Communications, Computers and signal Processing, 2005 IEEE Pacific Rim Conference on*, Aug. 2006.
- [17] Yao Zhao, Qiang Ma and Baozong Yuan, "Digital Watermark Based on LIFS," *Signal Processing Proceedings, WCCC-ICSP 2000. 5th International Conference on*, vol. 2, no. 3, Aug. 2000.
- [18] Ping-Sung Liao, Cheng-Yi Chen, Chun-Chi Chen, and Jeng-Shyang Pan, "Interlacing Domain Partition for Fractal Watermarking," *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP '06. International Conference on*, Dec. 2006.
- [19] Suman K. Mitra, C. A. Murthy and Malay K. Kundu, "Technique for fractal image compression using genetic algorithm," *IEEE Transactions on Image Processing*, vol. 7, pp. 586-593, April 1998.
- [20] W. H. Chen, "Fractal Image Compression using Ant Colony Optimization," *Department of Information Engineering, ISU*, June 2007.
- [21] G. Farhadi, "An Enhanced Fractal Image Compression Based on Quadtree Partition," *Proc. Int. Symp. Image and Signal Processing and Analysis*, pp. 213-216, 2003.
- [22] C.V. Serdean, M.K. Ibrahim, A. Moemeni, and M.M. Al-Akaidi, "Wavelet and multiwavelet watermarking," *Image Processing, IET*, vol. 1, June 2007.



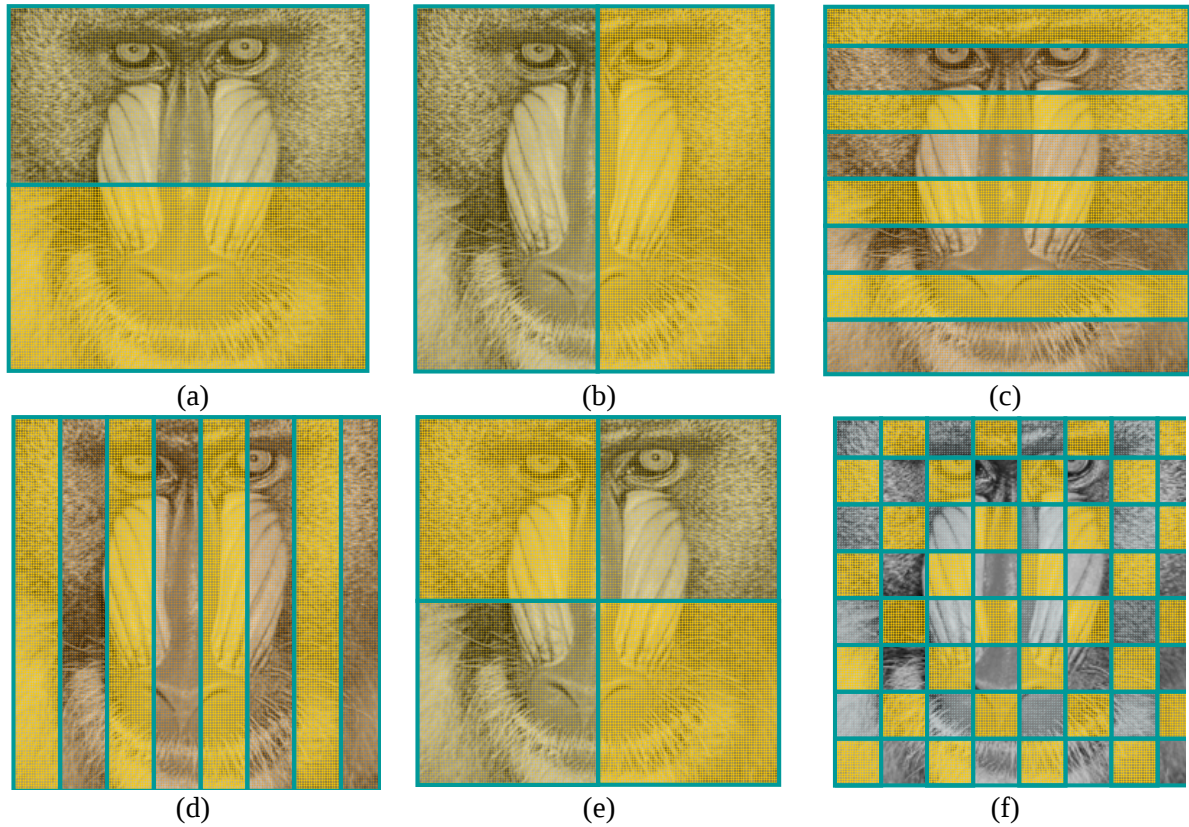
圖五、256×256 大小的影像解壓縮疊代過程：(a)原始影像，(b)初始影像，(c)疊代 1 次 (PSNR=15.10)，(d)疊代 2 次 (PSNR=17.42)，(e)疊代 3 次 (PSNR=21.75)，(f)疊代 5 次 (PSNR=30.96)，(g)疊代 7 次 (PSNR=33.63)，(h)疊代 12 次 (PSNR=34.51)，(i)疊代 20 次 (PSNR=34.56)



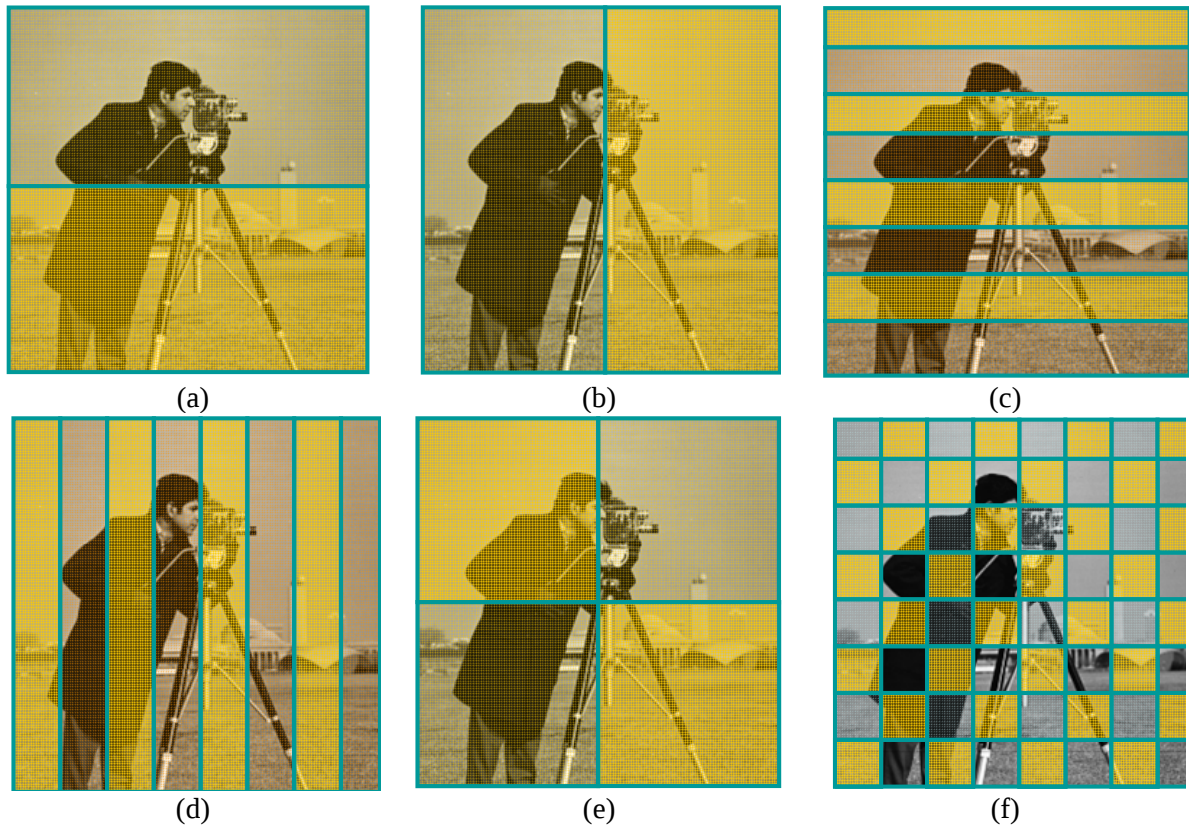
圖六、參數測試：(a)只搜尋左半邊 (PSNR=32.67)，(b)只搜尋上半部 (PSNR=33.52)，(c)只有四種旋轉角度 (PSNR=34.10)



圖七、位置參數測試：(a)PSNR=33.81，(b)PSNR=33.29，(c)PSNR=34.42，(d)PSNR=34.43，(e)PSNR=33.77，(f)PSNR=34.43



圖八、位置參數測試：(a)PSNR=27.18，(b)PSNR=27.21，(c)PSNR=27.56，(d)PSNR=27.56，(e)PSNR=27.23，(f)PSNR=27.56



圖九、位置參數測試：(a)PSNR=28.67，(b)PSNR=28.76，(c)PSNR=28.80，(d)PSNR=29.28，(e)PSNR=28.73，(f)PSNR=29.28