JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 17, 417-427 (2001)

Short Paper

Accelerating Fractal Compression With a Real-Time Decoder*

HSUEH-TING CHU AND CHAUR-CHIN CHEN
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
E-mail: cchen@cs.nthu.edu.tw

Image data compression by fractal techniques has been widely investigated. Although its high compression ratio and resolution-independent decoding properties are attractive, the encoding process is computationally demanding in order to achieve an optimal compression. A variety of speed-up algorithms have been proposed since Jacquin published a novel fractal coding algorithm. Unfortunately, the quantization strategy of scaling coefficients and the programming techniques lead to the results reported by different researchers are various even on the same image data which causes the speed-up of compression is incomparable. This paper proposes a real-time fractal decoder as a standard. We report the implementation results of a nearly *optimal* encoding algorithm OPT on commonly used images: *Jet, Lenna, Mandrill*, and *Peppers* of size 512×512. An accelerating compression algorithm using maximum gradient MG is shown to be 1300 times faster than OPT with a slight drop of PSNR value when encoding a 512×512 image.

Keywords: IFS, maximum gradient, fast encoder, simple decoder, collage theorem

1. INTRODUCTION

Data compression plays an important role in image analysis and transmission. The goal of image compression is to reduce storage space and to save transmission time. As media communication grows, image data compression attracts an increasing interest. Traditional transform coding techniques such as Karhunen Loeve transform, Walsh-Hadamard transform, and singular value decomposition techniques [15] used in 1970s and 1980s have been gradually replaced by new techniques. In 1990s, image compression algorithms based on wavelet transforms [2, 9, 18, 20, 27, 28], discrete cosine transform [24], vector quantization [10, 19, 22, 23, 26], and fractal approaches [1, 3-5, 12, 13, 25, 29] are widely investigated. A comparison of their advantages and disadvantages can be

---

Received June 2, 1999; revised March 15, 2000; accepted May 2, 2000.
Communicated by Zen Chen.
*This work was partially supported by NSC 88-2213-E-007-051.

can be found in [7]. This paper reports fractal-based image compression algorithms. The goal is to reveal the characteristics of small image blocks like 8×8 and to establish a simple fractal decoder and tries to build a simple fractal encoder.

Fractal encoding based on iterated function system (IFS) [1, 5, 11, 13] has been an active research field since the late 1980s. The main advantage of IFS-based fractal encoding is to avoid blockiness in achieving a high compression ratio by using the resolution-independent decoding property. Although fractal compression is an attractive technique, it faces some serious problems. First, fractal encoding is slow due to the tremendous amount of computations. Second, the existence problem of perfect domain-range matches in a fractal coding has not been proved yet. Third, the performance need be compared with a recently issued state-of-the-art compression algorithm based on wavelet transform [27, 28, 34].

This article reports experimental results on commonly used gray level images: *Jet, Lenna, Mandrill, and Peppers* by an optimal fractal compression algorithm and a simple accelerating fractal compression algorithm. The organization of this paper is as follows. Section 2 reviews mathematical foundation of fractal coding. Section 3 discusses a restricted optimal encoding algorithm. Section 4 introduces accelerating fractal compression algorithms. Section 5 demonstrates experimental results. Section 6 gives the conclusion.

2. MATHEMATICAL FOUNDATION

A fractal compression algorithm is mainly based on the famous Collage theorem and the fixed point theorem defined on a Banach space with the Hausdorff metric. We follow Barnsley's notations [5].

The Collage Theorem [5]: Let $\mathbf{Z}$ be a Banach space. Let $L \in H(\mathbf{Z})$ be given, where $H(\mathbf{Z})$ is the set consisting of all nonempty compact subsets of $\mathbf{Z}$. Let $\varepsilon \geq 0$ be arbitrarily chosen. Choose an iterated function system (IFS) of affine transformations $\{\mathbf{Z}; w_1, w_2, \ldots, w_N\}$ with contractivity factor $0 \leq s < 1$ so that

$$h(L, \bigcup_{i=1}^{N} w_i(L)) \leq \varepsilon, \tag{1}$$

where $h(\cdot)$ is the Hausdorff metric. Then

$$h(L, A) \leq \varepsilon, \tag{2}$$

where A is the attractor of the IFS.

This theorem indicates that given a compact set L, if we can find an IFS, $\{w_1, w_2, \ldots, w_N\}$ such that the union, or collage, $\bigcup_{i=1}^{N} w_i(L)$ is close to L, then the attractor of this IFS approximates L. In other words, L can be represented by this IFS [5].

For the application of the Collage theorem in image compression, we follow Jacquin's notations [13]. Denote $\mathbf{Z}$ as a set of all interesting images, and define a domain block D and a range block R as subimages of an element in $\mathbf{Z}$ such that the size of D is larger than that of R. In this paper, consider $\mathbf{Z}$ as the set of all 512×512 images with 256 possible gray levels, then $\mathbf{Z}$ can be regarded as the collection of all ordered 3-tuple points of the form $[x, y, f(x, y)]^t$, where $0 \le x, y \le 511$ and $0 \le f(x, y) \le 255$.

Definition: Let $\mathbf{Z}$ be a Banach space, and let $w_i: D_j \to R_i$ be a local contraction mapping on $\mathbf{Z}$ with the contractivity factor s_i for $1 \le i \le N$. Then

$$\tau = \{w_i: D_j \to R_i \mid 1 \le i \le N\} \tag{3}$$

is called a local IFS. The number $s = max\{s_i\}$ is called the contractivity factor of this local IFS if $0 \le s < 1$. Let $A, B \in \mathbf{Z}$, $D_j \subseteq A$, and $R_i \subseteq B$, then a local IFS can be written as

$$\tau: A \to B \text{ with } w_i: D_j \cap A \to R_i \cap B. \tag{4}$$

Thus, a local IFS can be regarded as a mapping on the image space $\mathbf{Z}$. The reason why fractal encoding/decoding works refers to [3, 13].

In the application of image data compression for $F \in \mathbf{Z}$, we try to find a contraction mapping, a local contractive IFS $\tau = \bigcup_{i=1}^{N} w_i$ with F the fixed point of τ. Some questions raised during the search of τ will be discussed in the following sections.

2.1 Fractal Encoding System

For a fractal encoding, we first partition an image into nonoverlapping $m \times m$ *range* blocks (m = 8 in this paper), $\{R_i\}$, for each R_i, we look for a 16×16 *domain* block which is closest to R_i in the sense of mean square errors. Suppose that there are N range blocks, we want to find a local IFS, τ, consisting of a set of restricted local affine transformations $\{w_i\}$ such that

$$\tau = \bigcup_{i=1}^{N} w_i, \qquad w_i : D_j \to R_i, \tag{5}$$

where w_i is called a restricted local affine transformation with the union of range blocks filling the whole image. In practical applications, we partition a digital image into nonoverlapping square range blocks R_i of size 8×8. Each range block is encoded by a local affine transformation w_i such that $R_i \approx w_i(D_j)$. In this paper, we consider 512×512 images with 16×16 domain blocks and 8×8 range blocks. Thus, a compressed code of an image consists of $\frac{512}{8} \times \frac{512}{8} = 4096$ restricted local affine transformations.

2.2.1 Restricted local affine transformations

A restricted local affine transformation is a composite of three local image operators which has the form

$$w = \psi \circ \pi \circ \xi., \tag{6}$$

where ξ, π and ψ are called spatial contraction, pixel permutation and block processing, respectively, which are described as follows.

Spatial contraction ξ: ξ shrinks a domain block to match the size of a range block and then translates the domain block to the position of the desired range block.

Pixel permutation π: π performs a rotation and/or a reflection on a 8×8 block $\xi(D)$, where is one of the following eight isometries [13].

$$\pi_0 = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}, \quad \pi_1 = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}, \quad \pi_2 = \begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}, \quad \pi_3 = \begin{bmatrix} 4 & 1 \\ 3 & 2 \end{bmatrix},$$

$$\pi_4 = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}, \quad \pi_5 = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}, \quad \pi_6 = \begin{bmatrix} 4 & 3 \\ 1 & 2 \end{bmatrix}, \quad \pi_7 = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}.$$

Block processing ψ: ψ performs as $\psi(\pi(\xi(D))) = s\cdot\pi(\xi(D)) + \mu 1$.

How to efficiently find an optimal local IFS τ such that $\tau(X) \approx X$ for a given image X is the central topic of fractal image coding which is discussed in the following sections.

2.2 A Fractal Decoding Algorithm

A simple decoding algorithm is listed below.

A Decoding Algorithm

1. Input a local IFS $\tau = \bigcup w_i$ and an image B.
2. Calculate $C = \tau(B)$.
3. $B \leftarrow C$.
4. Repeat Steps 2~3 K (K = 6 is used here) iterations until $C \approx B$.
5. B is the decoded image.

3. A NEARLY OPTIMAL IFS

To find a nearly optimal transformation w_i for a given range block R_i, we must search among all possible transformed domain blocks $\pi_k(\xi(D_{x,y}))$, $k = 0, 1, \ldots, 7$, $x, y = 0, 2, \ldots, 496$ to minimize $d(R_i, \psi(\pi_k(\xi(D_{x,y}))))$ and record (x, y, μ, k, s). The details are described as follows. Given a range block R, and a transformed domain block $\pi(\xi(D))$, let $a_j \in \pi(\xi(D))$ and $b_j \in R_j$, $0 \le j \le 63$. We seek s and μ to minimize the distortion measure

$$Error = \sum_{j=0}^{63}(s\cdot a_j + \mu - b_j)^2 = \sum_{j=0}^{63}\in_i^2. \tag{7}$$

The least squares error solution gives

$$s = \frac{n(\sum_{j=0}^{n-1} a_j b_j) - (\sum_{j=0}^{n-1} a_j)(\sum_{j=0}^{n-1} b_j)}{n\sum_{j=0}^{n-1} a_j^2 - (\sum_{j=0}^{n-1} a_j)^2}$$

$$\mu = \frac{\sum_{j=0}^{n-1} b_j - s\sum_{j=0}^{n-1} a_j}{n}, \qquad where \quad n = 64. \tag{8}$$

If $\left[n\sum_{j=0}^{n-1} a_j^2 - (\sum_{j=0}^{n-1} a_j)^2\right] = 0$, then $s = 0$ and $\mu = (\sum_{j=0}^{n-1} b_j)/n$.

To reduce the computations, the transformation ψ can be written [11] as

$$\psi(\hat{D}) = s \cdot (\hat{D} - U_d) + U_r$$
$$= (s \cdot c_0 + U_r, s \cdot c_1 + U_r, \ldots, s \cdot c_{n-1} + U_r)^t$$
$$where \quad c_j = a_j - U_d, \qquad j = 0,1,\ldots,n-1$$

where $\hat{D}$ is $\pi(\xi(D))$, D is a domain block,
U_d is the mean gray-level of the domain block,
U_r is the mean gray-level of the range block.

Thus, the solution for s and μ can be calculated by

$$s = \frac{\sum_{j=0}^{n-1} c_j b_j}{\sum_{j=0}^{n-1} c_j^2} \tag{10}$$
$$\mu = (\sum_{j=0}^{n-1} b_j)/n, \qquad where \quad n = 64.$$

Consider a fractal encoding for a 512×512 gray level image. We partition the image into 64×64 = 4096 nonoverlapping 8×8 *range* blocks, and partition the same image into 249×249 = 62001 overlapped 16×16 *domain* blocks by a spacing of 2 pixels along both row and column. In a restricted Jacquin's optimal fractal encoding algorithm (OPT) [13], for each range block R, we have to search a shrunken domain block $\xi(D)$ (to match the size of a range block) from the 8×62001 = 496008 possibilities with the *minimum* distortion measure which is computed by the square error between R and the transformed domain block $\pi(\xi(D))$ by

$$\left\| s \cdot \pi_k(\xi(D)) + \mu 1 - R \right\|_2^2, \quad \text{where } \pi_k \text{ is one of 8 possible isometrics [13],} \tag{11}$$

and the coefficients s and μ can be estimated by Eq. 8. For each match, one has to compute at least 5×64 = 320 floating-point operations. To encode a 512×512 image using the OPT algorithm, we require 320×4096×496001 operations to find the minimum distortions of all domain-range matches which is extremely time consuming. Any accelerating compressor tries to reduce the encoding time by finding a suboptimal match but lose the quality of a decoded image as little as possible. A quadtree partition using pattern recognition and clustering techniques is commonly used [12]. However, the quadtree approach uses various sizes of domain and range blocks whose speed-up and performance are incomparable with the OPT algorithm using fixed sizes of domain and range blocks. Another inconvenience of using quadtree approach is that the encoded image may have various number of bytes or bits to represent each range block which might require a more complex decoder or the decoder used for one image may not be used for other images which is impractical for current internet browsing applications. An OPT algorithm is listed below.

Algorithm OPT

1. Partition an image into nonoverlapping 8×8 range blocks.
2. Shrink the encoded image in half and collect all 8×8 blocks (maybe overlapped); these blocks associated with their 8 isometric transforms [13] establish the domain pool.
3. For each range block, find an optimal (shrunken) domain block in the domain pool with a minimum square error defined in Eq. 7. Record the location and the associated isometry of the optimal domain block, the quantized contractivity coefficient *s* and the quantized offset μ.
4. The encoded image is represented by a set of restricted local affine transformations (4096 for our applications), each transformation is represented by the location and one of the eight corresponding isometries of an optimal domain block, and quantized values of *s* and μ as (x, y, μ, k, s), where *s* is quantized using 5 bits by

$$s = (s < -1 ?\ 0 : s \geq 2.10 ?\ 31 : (int)\ (10.5 + s*10)) \tag{12}$$

The encoding process is computationally demanding by an exhaustive search of an optimal domain block for any given range block. Although Lee and Lee [16], and Lee and Ra [17] proposed using local variance to reduce the domain block search time for a given range block, they encountered the same problem of estimating the contractivity coefficients $\{s_i\}$. For various images, the range of the contractivity coefficients may be large enough to weaken the decoding fidelity. Lee and Lee's affine transforms set $s \equiv 1$. Lee and Ra assigns a few fixed values for the coefficients $\{s_i\}$. The next section proposes a *simple* approach to fast search for a suboptimal domain block for a given range block. Experiments are given in the following section.

4. ACCELERATING FRACTAL COMPRESSION ALGORITHMS

Initial experiments showed that the domain block and a range block with an optimal match essentially have the same location of peak intensity changes [ ]. This evidence can be briefly interpreted as follows.

From the equation (9), let $\epsilon_j = sa_j + \mu - b_j$. Then $s = \frac{b_j - \mu + \epsilon_j}{a_j}$ and $|\epsilon_j| \leq \sqrt{Error}$. If a domain block *D* is affine-similar to a range block *R*, then *Error* must be very tiny, ideally 0. By assuming that $|\epsilon_j| << |b_j - \mu|$ for each *j*, we have $\frac{b_j - \mu}{a_j} \approx \frac{b_k - \mu}{a_k} \approx s$, and hence $\frac{b_j - b_k}{a_j - a_k} \approx s$. This indicates that the difference between neighboring pixel values for a range block must be proportional to the corresponding part of an affine-similar domain block. Thus motivated, our algorithm searches only the shrunken domain blocks whose maximum intensity change at a pixel is close to the corresponding location of a given range block with maximum intensity change (maximum gradient). The algorithm is listed below.

Algorithm MG

1. Partition the domain pool into M chunks $\{C_1, C_2, \ldots, C_M\}$, e.g., $M = 256$.
2. Compute the maximum intensity change v_i from chunk C_i.
3. For each range block R_i, find the maximum intensity change u_i.
4. Find a domain-range block match, D_j, with $|u_i - v_j| \leq \Delta$, where v_j is the maximum intensity change of D_j, and Δ is a small integer, ideally 0.
5. Among the matching candidates, find a domain block which has the closest square distance, defined in Eq. 11, to the given range block.

The number of chunks, the Δ value, and the quantization of coefficient s corresponding to each range block will affect the performance of the Algorithm. We fix the number of chunks to be $M = 256$ with the chunk size 16×16, Δ is variable so that 256 candidates of domain blocks for each range block are all selected for a comparison to find a suboptimal restricted local affine transformation. It must be mentioned that if we set $M = 249\times249$ with the chunk size 1×1, Algorithm MG essentially performs like OPT. A comparison of the accelerating algorithm MG with OPT, and Lee [16] is given in the next section.

5. EXPERIMENTAL RESULTS

We tested a restricted optimal encoding algorithm (Algorithm OPT) and the proposed accelerating fractal compression Algorithm MG on four 512×512 images: *Jet*, *Lenna*, *Mandrill*, and *Peppers*. Images *Jet* and *Lenna* are shown in Fig. 1. All the experiments are done on a PC Celeron 333 running a Red Hat 5.2 Linux OS with 64M SDRAM. We fix the compression ratio to be 16 (0.5 bpp) without counting the potential reduction of lossless compression on quantized coefficients of affine transformations. The peak signal-to-noise ratio (PSNR) values between an original image f and the decoded image $\hat{f}$ associated with average encoding and decoding times are shown in Table 1.

(a) (b)

Fig. 1. Original 512×512 images: (a) Jet and (b) Lenna.

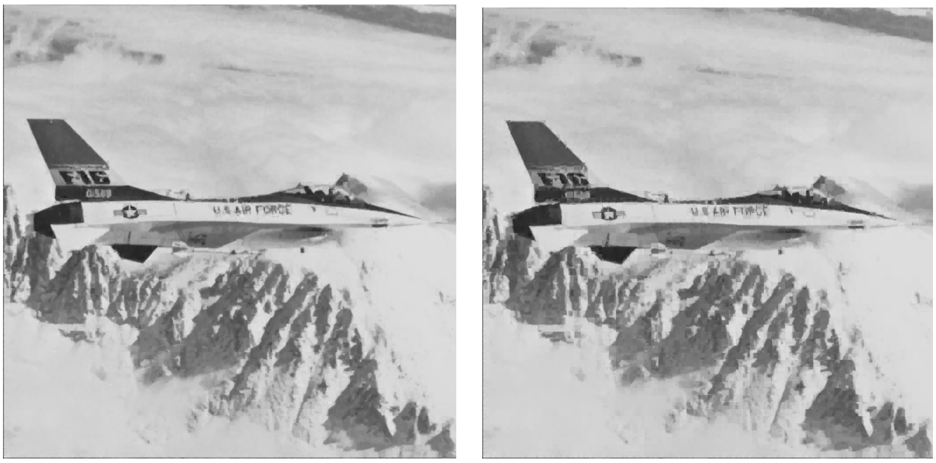
$$MSE = \frac{1}{MN} \sum_{I=0}^{M-1} \sum_{J=0}^{N-1} [\hat{f}(i, j) - f(i, j)]^2 \tag{13}$$

$$PSNR = 10 \log_{10} \left[\frac{255 \times 255}{MSE} \right] dB \tag{14}$$

Table 1. Performance of fractal coding algorithms on 512×512 images.

Algorithm	*PSNR values* (*in dB*)				*CPU time*	
	Jet	Lenna	Mandrill	Peppers	Encoding	Decoding
OPT	31.43	33.23	23.76	33.53	1300 min	1.3 sec
MG	29.20	31.27	22.58	31.64	1 min	1.3 sec
Lee [16]	26.47	29.46	21.76	29.05	2 min	1.3 sec

The corresponding decoded images for Algorithms OPT and MG are shown in Figs. 2-3, respectively. The experimental results show that fractal-based compression algorithms has PSNR values larger than those by VQ [7, 19]. Our proposed algorithms drop 2 dB of PSNR values while preserving the quality of decoding images at a certain level with a speed-up of over 1300 times of the nearly optimal algorithm OPT.



(a) (b)

Fig. 2. Decoded images of Jet by Algorithm (a) OPT and (b) MG.

We summarize our results that for an image containing a large of highly textured blocks, for example *Mandrill*, the PSNR values of all algorithms are significantly lower than those of images with a large portion of smooth regions such as *Lenna* and *Peppers*. Image *Jet* contains a certain portion of sharp edges and textures, so its PSNR values are between those of *Mandrill* and *Peppers*. The decoded images of Algorithms OPT and MG are well recognized and are satisfactory for all of the images. Although the PSNR values of OPT and MG algorithms for the image *Mandrill* are small, the decoded images are well recognized.

(a) (b)

Fig. 3. Decoded images of Lenna by Algorithm (a) OPT and (b) MG.

6. CONCLUSIONS

The fixed point theorem for a contractive local iterated function system has provided a good framework for image compression [5, 12, 13]. Its most attractive feature is the resolution-independent decoding property. However, the encoding is extremely time consuming. This report proposes a simple accelerating fractal compression algorithm using maximum gradient, the maximum intensity changes of neighboring pixels. Experimental results show that our algorithm is over 1300 times faster than a restricted optimal one which exhaustively searches many millions of domain blocks (4096×8×62001). Although our PSNR values may drop 2 dB compared with those of a nearly optimal algorithm, the proposed algorithm preserve visual quality at a certain level. Other features between domain and range blocks with optimal matches merit further investigation for more precise and faster encoding.

On the other hand, other accelerating fractal compression algorithms such as [6, 16, 17] may use different decoders for different fractal encoded images which may not be practical for the internet browsing-intensive applications. Instead, we propose a nearly real-time standard fractal decoder using 8×8 range blocks. Our decoder takes 1.3 seconds to decode a 512×512 compressed fractal image.

REFERENCES

1. L. F. Anson, "Fractal image compression," *Byte*, October, 1993, pp. 195-202.
2. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, Vol. 1, No. 2, 1992, pp. 205-220.
3. M. F. Barnsley, *Fractals Everywhere*, Academic Press, Boston, 1988.
4. M. F. Barnsley and A. D. Sloan, "A better way to compress images," *Byte*, January,

1988, pp. 215-223.
5. M. F. Barnsley and L. P. Hurd, *Fractal Image Compression*, AK Peters, Ltd. Wellesley, Massachusetts, 1993.
6. H. T. Chang and C. J. Kuo, "Adaptive schemes for improving fractal block coding of images," *Journal of Information Science and Engineering*, Vol. 15, No. 1, 1999, pp. 11-25.
7. C. C. Chen, "On the selection of image compression algorithms," in *Proceedings of IEEE International Conference on Pattern Recognition*, Vol. 2, 1998, pp. 1500-1504.
8. H. S. Chu, "A very fast fractal compression algorithm," M.S. Thesis, National Tsing Hua University, 1997.
9. I. Daubechies, Ten Lectures on Wavelet Analysis, *SIAM*, *J. Math. Analy.*, 1992.
10. W. H. Equitz, "A new vector quantization clustering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 10, 1989, pp. 1568-1575.
11. Y. Fisher, *Fractal Image Compression*, SIGGRAPH Course Notes, 1992.
12. Y. Fisher, Editor, *Fractal Image Compression: Theory and Applications*, Springer-Verlag, 1994.
13. A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, Vol. 1, No. 1, 1992, pp. 18-30.
14. A. E. Jacquin, "An introduction to fractals and their applications in electrical engineering," *Journal of the Fracklin Institute*, 1994, pp. 659-680.
15. A. K. Jain, *Digital Image Processing and Analysis*, Johns Wiley, 1989.
16. C. K. Lee and W. K. Lee, "Fast fractal image block coding based on local variances," *IEEE Transactions on Image Processing*, Vol. 7, No. 6, 1998, pp. 888-891.
17. S. M. Lee and S. W. Ra, "Fast fractal encoding algorithm using variances of image blocks," *Electronics Letters*, Vol. 34, No. 10, 1998, pp. 971-973.
18. A. S. Lewis and K. Knowles, "Image compression using 2D wavelet transform," *IEEE Transactions on Image Processing*, Vol. 1, No. 2, 1992, pp. 244-250.
19. Y. Linde, A. Buzo, and R. M Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, Vol. 36, No. 1, 1980, pp. 84-95.
20. S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, 1989, pp. 674-693.
21. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman & Company, 1983.
22. N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," *IEEE Transactions on Communications*, Vol. 36, 1988, pp. 957-571.
23. J. S. Pan, F. R. McInnes, and M. A. Jack, "Fast clustering algorithms for VQ," *Pattern Recognition*, Vol. 29, No. 3, 1996, pp. 511-518.
24. W. B. Pennebaker, J. Mitchell, *JPEG Still Image Compression Standard*, New York: Van Nostrand Reinhold, 1993.
25. D. C. Popescu, A. Dimca, and H. Yan, A nonlinear model for fractal image coding, *IEEE Transactions on Image Processing*, Vol. 6, No. 4, 1997, pp. 373-382.
26. B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Transactions on Communications*, Vol. 34, No. 11, 1986, pp. 1105-1115.
27. A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for*

Video Technology, Vol. 6, No. 3, 1996, pp. 243-250.
28. J. M. Shapiro, “Embedded image coding using zerotree of wavelet coefficients,” *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, 1993, pp. 3445-3462.
29. C. J. Wein and I. F. Blake, “On the performance of fractal compression with clustering,” *IEEE Transactions on Image Processing*, Vol. 5, No. 3, 1996, pp. 522-526.
30. ftp.uu.net:/graphics/jpeg/jpegsrc.v6a.tar.gz
31. http://www.mpeg.org/MPEG
32. ipl.rpi.edu/pub/EZ_Code/SPIHT
33. http://links.uwaterloo.ca
34. http://www.amara.com/current/Wavelet.html

Hsueh-Ting Chu (朱學亭) was born in Miaoli, Taiwan, on July 10, 1969. He received the B.S. degree in Computer and Information Science from Tung Hai University, Taichung, Taiwan, in 1991, and the M.S. degree in Computer Science form Naitonal Tsing Hua University, Hsinchu, Taiwan, in 1997. He is currently working toward the Ph.D. degree in Computer Science at National Tsing Hua University.

From 1991 to 1993, he was a corporal in army. After civil duty of army, he past the Senior Civil Service Examination, and worked as a computer engineer in Aerospace Science and Technology Research Center (ASTRC), National Cheng Kung University, Tainan, Taiwan from 1993 to 1995. His research interests include image processing and data compression.

Chaur-Chin Chen (陳朝欽) received the B.S. degree in mathematics from National Taiwan University, Taipei, in 1977, the M.S. degrees in both mathematics and computer science, and the Ph.D. degree in computer science, all from Michigan State University, East Lansing, Michigan, in 1982, 1984, and 1988, respectively. He worked as a research associate at Michigan State University and has been an associate professor in the Department of Computer Science at National Tsing Hua University, Hsinchu, Taiwan since August, 1989. He was a visiting scholar at Michigan State University from September to December in 1997. His research interests are in the areas of image pattern analysis and data compression.

Dr. Chen is a member of the IEEE Computer Society, the Association for Computing Machinery (ACM), the Pattern Recognition Society, and Image Processing and Pattern Recognition Society of Taiwan, R.O.C.