A New Affine Transformation: Its Theory and Application to Image Coding

Yao Zhao and Baozong Yuan, *Member, IEEE*

***Abstract*—The fractal image coding technique has attracted a degree of interest for its low bit rate. But the reconstructed image is of medium quality. This problem has prevented the fractal technique from being practically used. In order to improve the compression fidelity, a new affine transformation is proposed in this paper. Meanwhile, its contractivity requirement is analyzed, and the optimal parameters are derived using the least square method. The new affine transformation has been practically used in image coding. Experiments show that the PSNR can reach 28.7dB at a compression ratio (CR) of 16.4 for the 256 $\times$ 256 $\times$ 8 "Lena" image. Comparison with other fractal coding schemes shows that the new affine transformation can improve the reconstructed image quality efficiently.**

***Index Terms*—Affine transformation, digital impage compression, fractal image coding, iterated function system.**

I. Introduction

IT is in the past few years that fractal theory was applied in the field of image coding. In 1988, Barnsley and Sloan proposed using fractals generated by the (IFS) to encode or compress images [1]. But the first automatic robust block-based fractal image coding scheme which can compress any digital monochrome image was proposed by Jacquin in 1990 [2]. After that, some other papers which are related to fractal image coding were published [3]–[9]. But most of them are based on Jacquin's scheme.

However, experiments show that the fractal reconstructed image is of medium quality. The best result reported in [2] is that the PSNR = 27.7 dB and the bit rate = 0.68 bpp for 256 $\times$ 256 $\times$ 6 "Lena" image. In [2], some measures, such as the two-level partitioning technique, have been used to improve the fidelity. If only one-level partitioning is used, the PSNR will be even lower. The problem has attracted a degree of attention.

In this paper, a new affine transformation is proposed to improve the image quality. Its contractivity requirement is also analyzed, and the optimal parameters are derived. Finally, some experimental results and a comparison with other fractal coding schemes are presented.

II. The Mathematical Principle of Fractal Coding

Let (Ω, λ) denote a complete metric space. The elements of the space are digital images. λ is a given metric. The original image X_{orig} is one element of the space. The fractal coding procedure of X_{orig} is to construct a transformation $T: \Omega \rightarrow \Omega$, which satisfies the following conditions.

1) For any $p_1, p_2 \in \Omega$, $\lambda(T(p_1), T(p_2)) \leq s \cdot \lambda(p_1, p_2)$, where $0 \leq s < 1$.
2) $T(X_{\text{orig}}) \approx X_{\text{orig}}$.

From condition 1), we know that T is a contractive transformation. The *contractive mapping fixed point theorem* ensures that T has a unique fixed point, and the fixed point can be found by iteration of T. Condition 2) tells us that X_{orig} is an approximate fixed point of T. So X_{orig} can be reconstructed by applying T on any initial image X_0 iteratively. If T can be stored compactly, then it is called the compressed data of X_{orig}. Therefore, X_{orig} is compressed.

In practical use, it is difficult to construct such a T directly. It is usually constructed by the union of a series of contractive affine transformations.

$$T = \bigcup_{0 \leq i < N} T_i \tag{1}$$

where T_i is the ith contractive affine transformation, N is the total number of the affine transformations, and

$$T(X_{\text{orig}}) = \bigcup_{0 \leq i < N} T_i(D_i) \tag{2}$$

where $D_i \subset X_{\text{orig}}$.

The performance of fractal image coding mainly depends on the affine transformations. In the next section, we will propose a new affine transformation.

III. The New Affine Transformation

The new affine transformation we propose is

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & g(z) \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} e \\ f \\ 0 \end{bmatrix}. \tag{3}$$

(x, y) denotes the coordinates of a pixel to be transformed, z denotes the pixel intensity of (x, y), (X, Y) denotes the coordinates of the pixel transformed, Z denotes the intensity of (X, Y), and a, b, c, d, e, f are parameters of the affine transformation.

The affine transformation used in Jacquin's scheme is only a special case of (3) that $g(z) = tz + o$ [9]. Obviously, it is a linear one containing only a scaling and an offset. Its approximation ability is quite limited. The new affine transformation generalizes the pixel intensity approximation, so it can provide a better approximation.

Manuscript received January 13, 1995; revised September 28, 1995. This paper was recommended by Associate Editor K.-H. Tzou.

The authors are with the Institute of Information Science, Northern Jiaotong University, Beijing 100044, China.

Publisher Item Identifier S 1051-8215(98)02648-2.

For convenience of the following discussion, we rewrite (3) as (4)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & g(z) \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} e \\ f \\ 0 \end{bmatrix} \tag{4}$$

where

$$B = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

A. The Contractivity Condition of the New Affine Transformation

In order to make the decoding procedure converge, the affine transformation must be contractive. So we must discuss the contractivity condition of the new affine transformation. First, we introduce the definition of contractivity.

Definition 1: A transformation w is said to be contractive if, for any two points $\vec{x}_1$ and $\vec{x}_2$, the distance $\lambda(w(\vec{x}_1), w(\vec{x}_2)) \leq s \cdot \lambda(\vec{x}_1, \vec{x}_2)$ where $0 \leq s < 1$.

In this paper, for any vector $\vec{x} = (x_1, x_2, \cdots, x_n)^T$, the vector norm used is

$$\|\vec{x}\|_2 = \left(\sum x_i^2\right)^{1/2}$$

and the distance measure is

$$\lambda(\vec{x}_1, \vec{x}_2) = \|\vec{x}_1 - \vec{x}_2\|_2, \qquad \text{for any two vectors } \vec{x}_1, \vec{x}_2.$$

Next, we give a theorem about the contractivity of the new affine transformation.

Theorem 1: Suppose we already know that B is a contractive transformation (in a practical coding procedure, B maps a large block onto a small one, so the condition is usually satisfied); if $|g'(t)| \leq s$, $0 \leq s < 1$, then transformation (4) is a contractive transformation, where $g'(z)$ is the derivative of $g(z)$. $|g'(z)|$ is the absolute value of $g'(z)$

Proof: For any two vectors $\vec{x}_1 = (x_1, y_1, z_1)^T$, $\vec{x}_2 = (x_2, y_2, z_2)^T$, the transformed vectors are $\vec{X}_1 = (X_1, Y_1, Z_1)^T$, $\vec{X}_2 = (X_2, Y_2, Z_2)^T$. Then

$$\begin{aligned} \vec{X}_1 - \vec{X}_2 &= \begin{bmatrix} X_1 - X_2 \\ Y_1 - Y_2 \\ Z_1 - Z_2 \end{bmatrix} \\ &= \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & g(z_1) - g(z_2) \end{bmatrix} \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} B & 0 \\ 0 & g(z_1) - g(z_2) \end{bmatrix} \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \\ 1 \end{bmatrix}. \end{aligned}$$

For B to be a contractive transformation, according to the definition of contractivity, there exists an $s_1, 0 \leq s_1 < 1$, which satisfies

$$\begin{aligned} &\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \\ &\qquad \leq s_1 \cdot \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \end{aligned}$$

In terms of the Lagrange mean-value theorem,

$$\begin{aligned} Z_1 - Z_2 &= g(z_1) - g(z_2) \\ &= g'(\xi) \cdot (z_1 - z_2), \qquad \text{where } \xi \in (z_1, z_2). \end{aligned}$$

So

$$\begin{aligned} \sqrt{(Z_1 - Z_2)^2} &= |g'(\xi)| \cdot \sqrt{(z_1 - z_2)^2} \\ &\leq s \cdot \sqrt{(z_1 - z_2)^2}. \end{aligned}$$

Let

$$s_2 = \max\{s_1, s\}$$

so

$$\begin{aligned} &\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2} \\ &\qquad \leq s_2 \cdot \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}. \end{aligned}$$

Thus, the transformation is contractive. □

IV. Application of the New Affine Transformation to Image Coding

A. The Coding Scheme

Let X_{orig} be the original image. We assume that the size of X_{orig} is 256 × 256 × 8. Before compressing, we subtract 128 from all pixels of the image. This ensures that the sum total of the pixels is approximately zero. X_{orig} is first partitioned into 8 × 8 range blocks and 16 × 16 domain blocks which are denoted as $R_1, R_2, \cdots, R_i, \cdots, R_M$ and $D_1, D_2, \cdots, D_j, \cdots, D_N$, respectively. This is similar to [2], but only one-level partitioning is used in our experiments.

For any range block R_i, we search for a suitable affine transformation T_i and a domain block D_j to satisfy the following equation as well as possible.

$$R_i = T_i(D_j). \tag{5}$$

That is to say, we must find a suitable T_i and D_j to make $T_i(D_j)$ and R_i be similar to each other.

The affine transformation T_i maps a 16 × 16 domain block onto a 8 × 8 block, rotates or reflects the block, and processes the pixel intensity. All of the above functions are determined by the coefficients of the affine transformation T_i, but the coefficients are difficult to determine, quantize, and store. Usually, we replace T_i with a equivalent compound transformation:

$$T_i = G \circ \tau \circ \varphi. \tag{6}$$

So (5) changes to

$$R_i = G \circ \tau \circ \varphi(D_j) \tag{7}$$

where φ is the x–y plane contractivity transformation which maps D_j onto a 8 × 8 block, and τ is one of the eight rotation and flip operations proposed in [2]. G is the pixel intensity processor. Let z_i be the ith pixel intensity of $\tau \circ \varphi(D_j)$. Then the ith pixel intensity of $G \circ \tau \circ \varphi(D_j)$ is $g(z_i)$. Let Z_i denote the ith pixel intensity of R_i. Then the coding procedure of R_i is to select a suitable G, τ, D_j to minimize the following distortion:

$$\text{error} = \sum_{i=1}^{K} [Z_i - g(z_i)]^2 \tag{8}$$

where K is the total number of pixels in the range block.

When G, τ, and D_j are found, we store the parameters. Then the range block is encoded. When every range block is encoded in turn, the original image X_{orig} is encoded. Next, we will discuss some problems about the determination and storage of the parameters.

B. The Optimal Parameters

In transformation (3), $g(z)$ can be any form. In our experiments, $g(z) = \alpha_1 z + \alpha_2 z^2 + o$. So the distortion (8) changes to

$$\text{error} = \sum_{i=1}^{K} \left(Z_i - \alpha_1 z_i - \alpha_2 z_i^2 - o\right)^2. \tag{9}$$

The parameters α_1, α_2, o should minimize the distortion. Next, we use the least square method to determine the parameters. According to the least square method, we get the following equation group.

$$\begin{cases} \dfrac{\partial \text{error}}{\partial \alpha_1} = -2\displaystyle\sum_{i=1}^{K} \left(Z_i - \alpha_1 z_i - \alpha_2 z_i^2 - o\right) z_i = 0 \\ \dfrac{\partial \text{error}}{\partial \alpha_2} = -2\displaystyle\sum_{i=1}^{K} \left(Z_i - \alpha_1 z_i - \alpha_2 z_i^2 - o\right) z_i^2 = 0 \\ \dfrac{\partial error}{\partial o} = -2\displaystyle\sum_{i=1}^{K} \left(Z_i - \alpha_1 z_i - \alpha_2 z_i^2 - o\right) = 0. \end{cases} \tag{10}$$

Solving (10), we get the optimal parameters

$$\begin{cases} \alpha_1 = \dfrac{\left(\overline{Zz} - \overline{Z}\,\overline{z}\right)\left(\overline{z^2}^2 - \overline{z^4}\right) - \left(\overline{Zz^2} - \overline{z^2}\,\overline{Z}\right)\left(\overline{z^2}\,\overline{z} - \overline{z^3}\right)}{\left(\overline{z^2}\,\overline{z} - \overline{z^3}\right)^2 - \left(\overline{z}^2 - \overline{z^2}\right)\left(\overline{z^2}^2 - \overline{z^4}\right)} \\ \alpha_2 = -\dfrac{\left(\overline{Zz} - \overline{Z}\,\overline{z}\right) + \alpha_1\left(\overline{z}^2 - \overline{z^2}\right)}{\overline{z^2}\,\overline{z} - \overline{z^3}} \\ o = \overline{Z} - \alpha_1 \overline{z} - \alpha_2 \overline{z^2} \end{cases} \tag{11}$$

where

$$\overline{Z} = \frac{1}{K}\sum_{i=1}^{K} Z_i, \quad \overline{z} = \frac{1}{K}\sum_{i=1}^{K} z_i, \quad \overline{z^2} = \frac{1}{K}\sum_{i=1}^{K} z_i^2,$$
$$\overline{z^3} = \frac{1}{K}\sum_{i=1}^{K} z_i^3, \quad \overline{z^4} = \frac{1}{K}\sum_{i=1}^{K} z_i^4, \quad \overline{Zz} = \frac{1}{K}\sum_{i=1}^{K} Z_i z_i,$$
$$\overline{Zz^2} = \frac{1}{K}\sum_{i=1}^{K} Z_i z_i^2.$$

After α_1, α_2, o are calculated, we must verify if the parameters satisfy the contractivity condition mentioned in Theorem 1. Sometimes, the parameters do not satisfy the contractivity condition.

According to Theorem 1, α_1, α_2, o should satisfy the following equation

$$|g'(z_i)| = |\alpha_1 + 2\alpha_2 z_i| < 1. \tag{12}$$

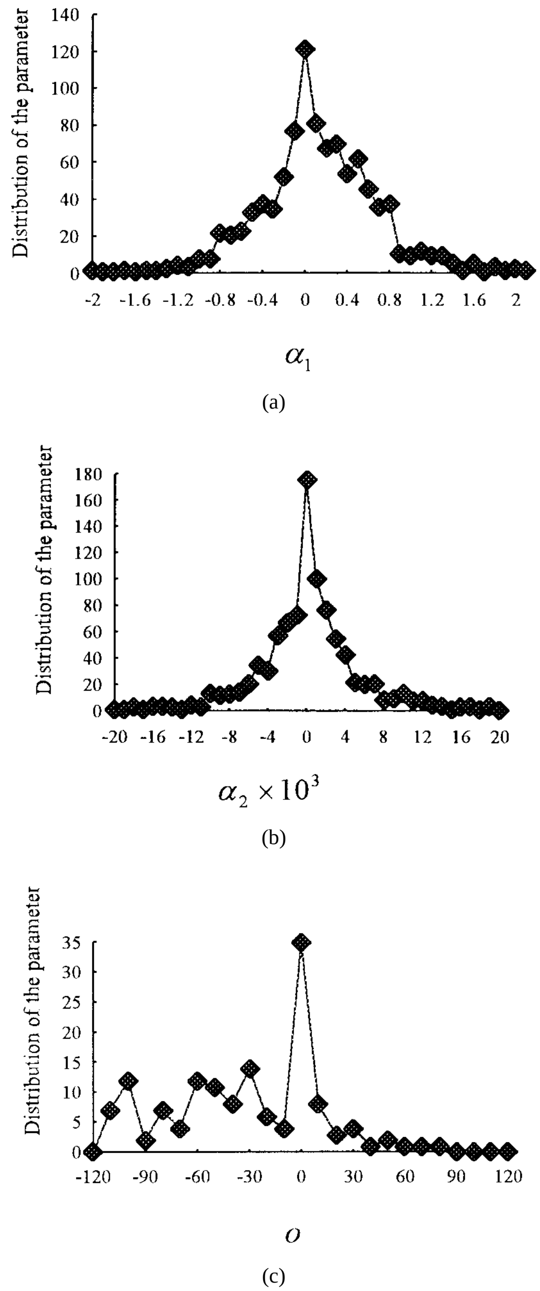

Fig. 1. (a) Distribution curve of α_1. (b) Distribution curve of α_2. (c) Distribution curve of o.

Therefore

$$-1 < \alpha_1 + 2\alpha_2 z_i < 1 \tag{13}$$

where z_i is the pixel intensity of the block $\tau \circ \varphi(D_j)$. Every pixel of the block $\tau \circ \varphi(D_j)$ must satisfy (13). If we examine every pixel, the computing is too complex. Analyzing (13), we find that it is equivalent to the following two equations:

$$-1 < \alpha_1 + 2\alpha_2 z_{\max} < 1 \tag{14}$$
$$-1 < \alpha_1 + 2\alpha_2 z_{\min} < 1 \tag{15}$$

where $z_{\max}$ and $z_{\min}$ are the maximum pixel and the minimum pixel of the block $\tau \circ \varphi(D_j)$, respectively.

Fig. 2. 256 × 256 × 8 original image "Lena."

Fig. 3. Decoded image using the traditional method (one-level partitioning).

If α_1, α_2, o do not satisfy (14) and (15), the group of parameters cannot be used; we will compute the next optimal parameters, and examine their contractivity.

When the parameters are calculated and examined, we will quantize and store them.

C. The Quantization of the Parameters

When D_j, G, τ which satisfy (7) are found, we must quantize and store the parameters. The parameters needed to be stored are the position information of D_j, and the rotation and flip operations τ, α_1, α_2, o.

The bit allocations are as follows.

1) The position information of D_j.

The size of the original image is 256 × 256, and the neighboring domain blocks separate four pixels, so it needs $6 + 6 = 12$ bits to store the coordinates of the domain blocks D_j.

2) The rotation and flip operation τ.

There are eight kinds of rotation and flip operations, so 3 bits are needed to store it.

Fig. 4. Decoded image using the method presented in the paper (one-level partitioning).

α_1, α_2, o are difficult to quantize. Next, we discuss their quantization.

The distribution curves of α_1, α_2, o obtained from the compression of the "Lena" image are shown in Fig. 1(a)–(c).

3) From Fig. 1(a), we find that most of α_1 varies between −1.5 to 1.5. so we quantize it in the following way.

If $-1.55 < \alpha_1 < 1.55$, then $0 \leq \mathrm{QUAN}[(\alpha_1/0.1)] + 15 \leq 30$, where $\mathrm{QUAN}[\,]$ is a function which takes the integer nearest to its variable. Obviously, it needs 5 bits to store it.

If $\alpha_1 \geq 1.55$ or $\alpha_1 \leq -1.55$, then let $p[\alpha_1] = \mathrm{SIGN}[\alpha_1] \cdot [\mathrm{ABS}(\alpha_1) - 1.5]$, where

$$\mathrm{SIGN}[\alpha_1] = \begin{cases} 1, & \text{if } \alpha_1 \geq 0 \\ -1, & \text{if } \alpha_1 < 0. \end{cases}$$

$\mathrm{ABS}[\alpha_1]$ takes the absolute value of α_1. Then, as for $p[\alpha_1]$, its distribution property is similar to α_1. So we can use the above method to quantize it, but this time, the flag code $(1111)_2$ should be placed forward.

4) From Fig. 1(b), we find that most α_2 satisfy

$$-15.5 < \frac{\alpha_2}{10^{-3}} < 15.5. \qquad (16)$$

So, we can use a similar method to quantize and store α_2.

5) From Fig. 1(c), we find that most o distribute between −128 to 127. So if $-128.5 < o < 126.5$, then

$$0 \leq \mathrm{QUAN}[o] + 128 < 254. \qquad (17)$$

This needs 8 bits to store it. If $o \leq -128.5$ or $o \geq 126.5$, we can use a similar method to 3), but the flag code $(11111111)_2$ should be placed forward.

V. Experimental Results

Experiment 1: The first experiment is used to test if the new affine transformation can improve the image quality. The original image used is the 256 × 256 × 8 standard image "Lena"

Fig. 5. Unconverged reconstructed image sequence.

which is shown in Fig. 2. We have compressed the original image using both the traditional method [2] and the method proposed in the paper. In this experiment, only a one-level partitioning technique is used in both schemes. The size of the range block is 8 × 8, and the domain block size is 16 × 16. We divide the range blocks into four kinds to process them differently as in [2]. The compression ratio (CR) and peak-to-peak signal-to-noise ratio (PSNR) are chosen as criteria of comparison. The result is as follows.

When using the traditional method, the compression ratio CR $= 17.8$, the PSNR $= 24.9$ dB.

When using the method proposed in the paper, the compression ratio CR $= 16.4$, and the PSNR $= 28.7$ dB.

The decoded images are shown in Figs. 3 and 4.

Comparing Figs. 3 and 4, we can easily find that the fidelity of Fig. 4 is better than that of Fig. 3, especially at the edge of the column in the image. Comparing the PSNR and CR indicates that the PSNR can increase nearly 4 dB, while the CR decreases a little. So the new affine transformation can improve the image quality.

One important point must be mentioned in the decoding procedure. For the case where the affine transformation is linear, the contractivity condition does not depend on the gray level of the domain pixels; therefore, any initial image can be used as the starting point of the decoding process. The new affine transformation discussed in this paper has a second-order term; therefore, the contractivity of each transformation does depend on the gray level of the pixels in the domain blocks, as shown in (14) and (15). Therefore, the convergence of the decoding scheme depends on the initial image used as the starting point of the decoding process. In this experiment, when a black image whose gray is near zero is chosen as the initial image, the decoding can converge, but when the gray is near 127 or -128 (because, in the encoding procedure, the original image has been subtracted 128 from all pixels), the decoding does not. The unconverged reconstruction sequence is shown in Fig. 5. We find that there are many white and black points in the iteration image, and as the transformations iterate, the number of white and black points increases quickly, and the decoding image becomes disorderly. This problem affects the practical usage of the new affine transform.

Fig. 6. Decoded image using the traditional method (two-level partitioning).

But we found that when a limitation is applied after each iteration, the decoding can be made to converge. The limitation used was the following.

After each iteration, the gray of the iteration image is limited within -128 and 127. That is,

$$\begin{aligned}&\text{if } z_i > 127, \quad \text{let } z_i = 127;\\&\text{if } z_i < -128, \quad \text{let } z_i = -128;\\&\text{if } -128 \le z_i \le 127, \quad z_i \text{ remains unchanged}\end{aligned}$$

where z_i is the gray level of the iteration image.

Although this limitation was sufficient for convergence of the image tested, a more rigorous limitation would be to determine $z_{\max}$ and $z_{\min}$ from (14) and (15) for each transformation, and limit the gray values in the domains of each transformation accordingly.

Experiment 2: The experiment is also used to test the performance of the new affine transform. This time we use 256 × 256 × 6 "Lena" as the original image. We also compress the original image using both the traditional method and the method proposed in the paper. But in this experiment, we use the two-level partitioning technique which was proposed in [2].

Fig. 7. Decoded image using the method presented in the paper (two-level partitioning).

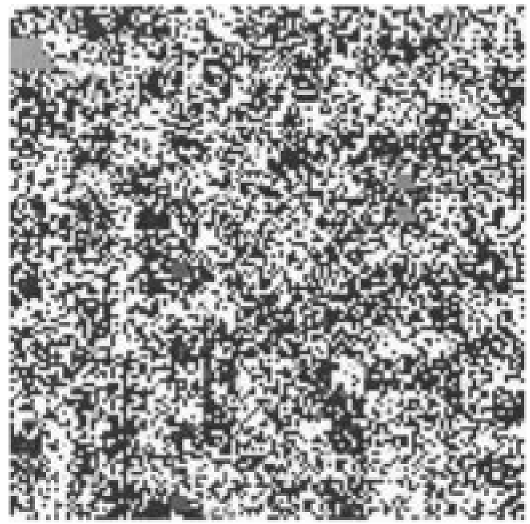

Fig. 8. Decoded image when the contractivity condition is not applied.

The range block sizes are 8×8 and 4×4, and the domain block sizes are 16×16 and 8×8. The results are as follows.

When using Jacquin's method, the compression ratio CR $= 8.5$, and the PSNR $= 28.3$ dB.

When using the method proposed in the paper, the compression ratio CR $= 8.4$, and the PSNR $= 31.7$ dB.

The decoding images are shown in Figs. 6 and 7.

Experiment 3: The third experiment is done to verify the contractivity condition theorem, Theorem 1. In the coding of the original image, for any range block, we must search for the most similar domain block and calculate the optimal parameters. As mentioned in Section IV-B, some parameters do not satisfy the contractivity condition. We have to examine the parameters and process them. So in order to test the efficiency of Theorem 1, we compress the original image in two different ways. First, we examine the parameters and process them to make every parameter satisfy the contractivity condition; the decoding procedure can converge to a good image. Then, if we omit the examination and processing procedure, the decoding procedure does not converge. Whether or not we use the limitation in Experiment 1, the decoded image is disorderly. It is shown in Fig. 8.

REFERENCES

[1] M. F. Barnsley and A. Sloan, "A better way to compress images," *BYTE Mag.*, Jan. 1992.

[2] A. E. Jacquin, "A novel fractal block-coding technique for digital images," in *Proc. ICASSP*, 1990, pp. 2225–2228.

[3] J. M. Beaumont, "Image data compressing using fractal techniques," *BT Technol. J*, vol. 9, no. 4, pp. 93–109, 1991.

[4] R. Dettmer, "Fractal-based image compression," *Inst. Elect. Eng. Rev.*, pp. 323–327, Sept. 1992.

[5] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: A study of the transform method," *Signal Process.*, no. 29, pp. 251–263, 1992.

[6] D. M. Monro and F. Dudbridge, "Fractal block coding of images," *Electron. Lett.*, vol. 28, no. 11, pp. 1053–1055, 1992.

[7] D. M. Monro, "Class of fractal transforms," *Electron. Lett.*, vol. 29, no. 4, pp. 362–363, 1993.

[8] Y. Zhao and B. Yuan, "Image compression using fractals and discrete cosine transform," *Electron. Lett.*, vol. 30, no. 6, pp. 474–475, 1994.

[9] G. Lu, "Fractal image compression," *Signal Process.: Image Commun.*, vol. 5, pp. 327–262, Oct. 1993.