Relocation, scaling, and quantization effects on fractal images

Hsuan T. Chang, MEMBER SPIE
Chaoyang University of Technology
Department of Information Management
Wufeng, Taichung, 413, Taiwan
E-mail: htchang@cyut.edu.tw

Abstract. We investigate some practical problems and propose their solutions for the display of fractal images. The original sizes and coordinates of fractal images are usually different, and they can be measured only from the decoded results. This is a drawback, since the decoded fractal image may not be shown with adequate size and may not locate in a desired position. In this paper, we propose a hierarchical fixed-point-seeking method that can directly determine the original size and the coordinates of a fractal image from its iterated function system (IFS) code. With a modification of the translation parameters in the IFS code, we can resize and relocate the decoded image. On the other hand, we discuss the quantization effects due to the finite word length in the hardware implementation. Finally, a method for determining the iteration number in the random iteration algorithm is also proposed. In computer simulation, we demonstrate some examples to verify the proposed solutions.
[DOI: 10.1117/1.1367345]

Subject terms: fractal image; hierarchical search; fixed point; iterated function system; random iteration algorithm.

Paper 200224 received June 6, 2000; revised manuscript received Dec. 29, 2000; accepted for publication Jan. 10, 2001.

1 Introduction

Fractal techniques for image coding have been greatly developed since their invention by Barnsley.[1] The basic idea originates from the observation that natural images such as clouds, mountains, coastlines, plants, etc., have high *self-similarity*. That is, these natural images have the property that their magnified subsets look like the whole set. Barnsley found that a finite set of specific contractive transform functions (CATs) of an iterated function system (IFS) can be used to generate a fractal image with a random iteration algorithm.[1–3] The IFS can achieve a very high compression ratio ($>$10,000), since only a few sets of parameters specifying the CATs are required. Its decoding algorithm is simple, but both the encoding and decoding processes are computation-intensive.

In the past decade, some methods[4–6] have been proposed to solve the inverse problem of the encoding procedure, and some parallel algorithms[7–9] have been provided to speed up the decoding procedure. On the other hand, some optical architectures[10–12] have been proposed and implemented to show the parallelism and high speed of which optics is capable. However, there have been no studies, to our knowledge, to investigate the scaling and relocation problems for the fractal images in the encoding stage. The original sizes of decoded fractal images are different, since the specification of the image size is not standardized in the encoding stage. Thus an image with an inadequate size may appear on the display. To normalize the image, we must measure the size of the decoded image and then proceed with resizing. This process is troublesome and not practical, especially for hardware implementation. Therefore, one should utilize some modification on the IFS codes *before* the fractal image is generated such that the decoded fractal images are of proper sizes.

In this paper, we propose a *hierarchical fixed-point-seeking* (HFPS) method that can efficiently determine the size of a fractal image from IFS codes. That is, the proposed HFPS method can determine the original size and coordinates of fractal image, which are then used to modify the original IFS code. This method originates from the Banach fixed-point theorem[1] and makes use of Pei's parallel algorithm.[8] First of all, we calculate the fixed points of all CATs and construct a contour that connects all of the points without overlap. The sublevel CATs of each CAT can be derived by Pei's parallel algorithm. By hierarchically calculating the fixed points of sublevel CATs, we can determine the final contour when no fixed points of the next-level CATs contribute the extension to the contour. Therefore, original size of the decoded fractal image is obtained directly from its IFS code. Second, we employ the coordinate transformation to modify the IFS code according to the original and the desired sizes. With the random iteration algorithm, the decoded fractal image can be generated with a desired size no matter what the original size is in the encoding stage. Third, the quantization effects due to the finite word length used in the computing process of the random iteration algorithm are also investigated for hardware implementation. Finally, the iteration number in the random iteration algorithm is discussed, so that we can decode a fractal image more efficiently.

The organization of this paper is as follows: Section 2 briefly reviews fractal image coding based on IFSs. In Sec. 3, we first describe the proposed HFPS method that can automatically determine the coordinate and the size of

Table 1 IFS codes for Sierpiński triangle, fern, castle, and snowflake.

Image	W	a	b	c	d	e	f	p
Sierpiński triangle	1	0.5	0	0	0.5	0	0	0.33
	2	0.5	0	0	0.5	1	0	0.33
	3	0.5	0	0	0.5	0.5	0.5	0.34
Fern	1	0	0	0	0.16	0	0	0.01
	2	0.2	−0.26	0.23	0.22	0	1.6	0.07
	3	−0.15	0.28	0.26	0.24	0	0.44	0.07
	4	0.85	0.04	−0.04	0.85	0	1.6	0.85
Castle	1	0.5	0	0	0.5	0	0	0.25
	2	0.5	0	0	0.5	2	0	0.25
	3	0.4	0	0	0.4	0	1	0.25
	4	0.5	0	0	0.5	2	1	0.25
Snowflake	1	0.333	0	0	0.333	0.333	0	0.2
	2	0.333	0	0	0.333	0	0.333	0.2
	3	0.333	0	0	0.333	0.333	0.333	0.2
	4	0.333	0	0	0.333	0.333	0.666	0.2
	5	0.333	0	0	0.333	0.666	0.333	0.2

original image. Then the IFS code is modified according to the measured information. Section 4 investigates the quantization effects due to the finite word length. Section 5 deals with the iteration number in the random iteration algorithm for fractal image decoding. The experimental results are given in Sec. 6. Finally, Sec. 7 concludes this paper.

2 Iterated Function Systems

In this section we review the IFS and the random iteration algorithm for fractal image decoding. The Banach fixed-point theorem and Pei's parallel algorithm are also briefly reviewed, since both are used in the proposed HFPS method.

2.1 *Contractive Affine Transformation*

IFS theory is an extension of classical geometry. It uses CATs to express relations between parts of an image. The general form of a CAT W for a point located at the position (x,y) is expressed by[2]

$$W\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \mathbf{A}\begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{b} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, \tag{1}$$

where the coefficients a, b, c, d, e, and f are real numbers and $|ad-bc|<1$. These coefficients are used to describe relationships of the rotations, scaling, and translations between the subimages and the whole image. Because fractal images have self-similarity, we may find N CATs, $\{W_i, i=1,2,\ldots,N\}$, such that

$$F = \bigcup_{i=1}^{N} W_i(F), \tag{2}$$

where F is a fractal image and $W_i(F)$ is a subimage given by the CAT W_i. Table 1 collects the IFS codes of the Sierpiński triangle, fern, castle, and snowflake.

2.2 *Random Iteration Algorithm*

There are two basic decoding algorithms for the IFS code to decode a fractal image: the deterministic algorithm and the random (probabilistic) iteration algorithm. The random iteration algorithm is the most frequently used, since it uses the property of probability corresponding to each CAT and hence the amount of computation can be reduced. Assume that the IFS code is known. The random iteration algorithm is summarized as follows:

1. An arbitrary initial point (x_t, y_t) is chosen.
2. For $t=0$ to #itr, do steps 3 to 5.
3. One of the CATs W_i, $i=1,\ldots,N$, is randomly selected with probability p_i.
4. Apply the transformation W_i to the (x_t, y_t) to obtain (x_{t+1}, y_{t+1}).
5. Plot (x_t, y_t).

In step 2, the symbol #itr indicates the iteration number of points. Let the iteration number be 9000. The decoded fractal images corresponding to the IFS codes in Table 1 are shown in Figs. 1(a) to 1(d). Clearly, different fractal images have different sizes and are located in different positions. On the other hand, because of their different contents, it is inefficient to choose the same iteration number in the random iteration algorithm. We give a solution to this problem in Sec. 5.

2.3 *Pei's Parallel Decoding Algorithm*

Pei proposed a parallel decoding algorithm[8] that can determine the fixed point of each CAT to solve the problem of

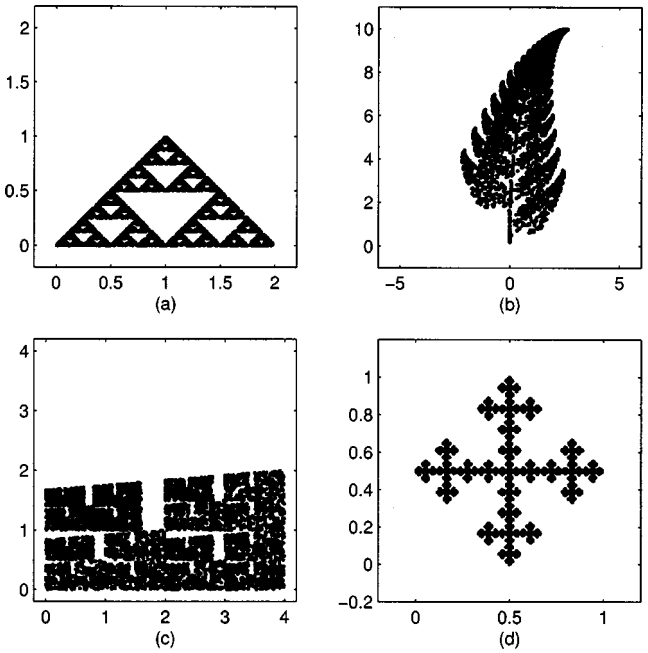

Fig. 1 The fractal images decoded by the random iteration algorithm: (a) Sierpiński triangle, (b) fern, (c) castle, and (d) snowflake.

the transient points. The block diagram of Pei's algorithm is shown in Fig. 2. In Pei's method, the fixed point g_i of the corresponding CAT W_i is calculated by

$$g_i=\begin{bmatrix}x_i\\ y_i\end{bmatrix}=(\mathbf{I}-\mathbf{A}_i)^{-1}\mathbf{b}_i\,. \tag{3}$$

The fixed point of the CAT W_i must be in the subset $W_i(F)$. First we calculate the fixed points of all CATs. These fixed points serve as the initial points to avoid transient behavior. In this figure, each CAT W_i is decomposed as

$$W_i=\bigcup_{j=1}^{N} T_{ij}\,, \tag{4}$$

where the sublevel CAT $T_{ij}=W_iW_jW_i^{-1}$, and therefore,

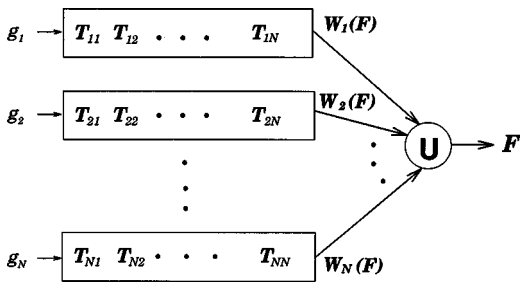

Fig. 2 Pei's parallel decoding architecture.

Table 2 The measured x and y ranges and the size of four fractal images: Sierpiński triangle, fern, castle, and snowflake.

Image	x_{min}	x_{max}	y_{min}	y_{max}	Size
Sierpiński triangle	0	2	0	1	2×1
Fern	−2.18	2.66	0	10.0	4.8×10
Castle	0	4	0	2	4×2
Snowflake	0	1	0	1	1×1

$$T_{ij}\left(\begin{bmatrix}x\\ y\end{bmatrix}\right)=\mathbf{A}_i\mathbf{A}_j\mathbf{A}_i^{-1}\begin{bmatrix}x\\ y\end{bmatrix}-\mathbf{A}_i\mathbf{A}_j\mathbf{A}_i^{-1}\mathbf{b}_i+\mathbf{A}_i\mathbf{b}_j+\mathbf{b}_i$$
$$=\mathbf{A}_{ij}\begin{bmatrix}x\\ y\end{bmatrix}+\mathbf{b}_{ij}\,. \tag{5}$$

Note that T_{ii} is exactly equal to W_i without requiring any computation. Therefore, we need to determine N^2-N new sublevel CATs. From the equation above, we obtain that

$$\mathbf{A}_{ij}=\mathbf{A}_i\mathbf{A}_j\mathbf{A}_i^{-1},\quad \mathbf{b}_{ij}=-\mathbf{A}_i\mathbf{A}_j\mathbf{A}_i^{-1}\mathbf{b}_i+\mathbf{A}_i\mathbf{b}_j+\mathbf{b}_i\,. \tag{6}$$

Since the fixed point of the CAT W_i can be calculated by Eq. (3), the fixed point g_{ij} of the sublevel CAT T_{ij} can be calculated in a similar way. The extension capability of Pei's algorithm shows that each sublevel CAT can be further decomposed into its sublevel CATs. By using Eq. (6), therefore, we can recursively extend the sublevel CATs of T_{ij}, T_{ijk}, and then the sublevel CATs of T_{ijk}, T_{ijkl}, and so on. For each extended sublevel CAT, we can determine the corresponding fixed point.

In the above algorithm, the inverse matrix $\mathbf{A}^{-1}$ may not exist when some of the coefficients in matrix $\mathbf{A}$ are zero. We then have to modify the coefficient zero to be a small positive number, for example, 0.00001. Then the inverse matrix in Eq. (6) exists, and Pei's algorithm can work properly.

2.4 *Problem Description*

Figure 1 shows the fractal images decoded by the random iteration algorithm. The original sizes of these images can be obtained from the maximum and the minimum of the x and y coordinates in all generated pixels. Table 2 shows the x and y ranges and the sizes of four decoded fractal images: Sierpiński triangle, fern, castle, and snowflake, respectively. Their sizes are different, and thus the scaling problem arises if we hope to show these images with the same size. For example, Fig. 3 shows the decoded results if the range of display is chosen as 3×3. A display with a fixed size may result in a decoded fractal image that is too small for us to observe [Fig. 3(d)], it may not locate it in a desirable position [Fig. 3(a)], or only part of the image may be presented [Figs. 3(b) and 3(c)]. Because we cannot determine their sizes and their coordinates in advance, both shifting and scaling of the decoded image are necessary so that we can obtain an image with a desired size and in a desired position. However, the postprocessing above is troublesome. First of all, we must measure the maximum and minimum values of the x and y coordinates of all pix-

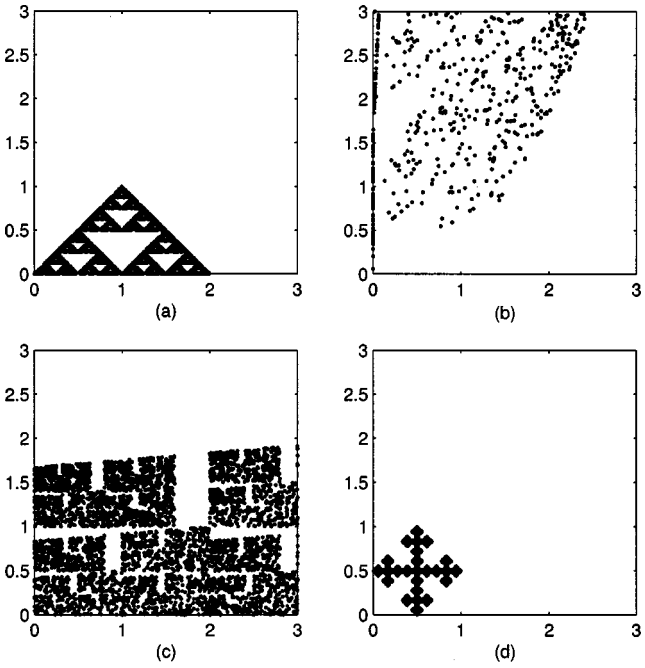

Fig. 3 The decoded fractal images within the same range $\{(x,y)|0\leqslant x\leqslant 3, 0\leqslant y\leqslant 3\}$: (a) Sierpiński triangle, (b) fern, (c) castle, and (d) snowflake.

els. Then, both the scaling and the translation operations must be applied to the entire image to obtain a rescaled image in a desired position.

In the following sections, we first propose a novel method that determines the image size and coordinates directly from the IFS code. The IFS code is then modified so that we can directly decode a desired size of fractal image in a desired location. The quantization effects due to the finite word length are also investigated. Finally, it is not reasonable to select the same iteration number in the random iteration algorithm for every image. A selection criterion for the iteration number is also provided in this paper.

3 IFS Code Modification for Scaling and Relocation

In this section, we propose a *hierarchical fixed-point-seeking* (HFPS) method to automatically determine the original size and coordinates of a fractal image. Then we modify the translation parameters in the IFS code. The fractal image can thus be generated with a desired size and at a desired location.

3.1 *HFPS Method*

Figure 4 shows a diagram of the proposed HFPS method. First of all, the fixed point g_i for each CAT W_i is calculated. Then all fixed points are sequentially connected to obtain a contour. This contour serves as the level-one contour. For each CAT W_i, we then find its child transformations $T_{ij}^{(2)}$ and the corresponding fixed points $g_{ij}^{(2)}$. Similarly, we plot the level-two contour using the fixed points $g_{ij}^{(2)}$ and check whether the contour is extended. Once a fixed point $g_{ij}^{(2)}$ contributes the extension to the contour, we set it as the new contour point that will be used to determine the fixed points at the next level (level three). The process above is executed hierarchically and recursively until the criterion that no fixed point at level $n+1$ makes an extension of the contour at level n is satisfied. We thus obtain size and coordinate information on the fractal image from the x and y coordinates of four points located at the north, south, west, and east corners of the contour. Therefore the width and the length of the original fractal image are $x_{\max}-x_{\min}$ and $y_{\max}-y_{\min}$, respectively. The proposed HFPS method is summarized as the following pseudocode:

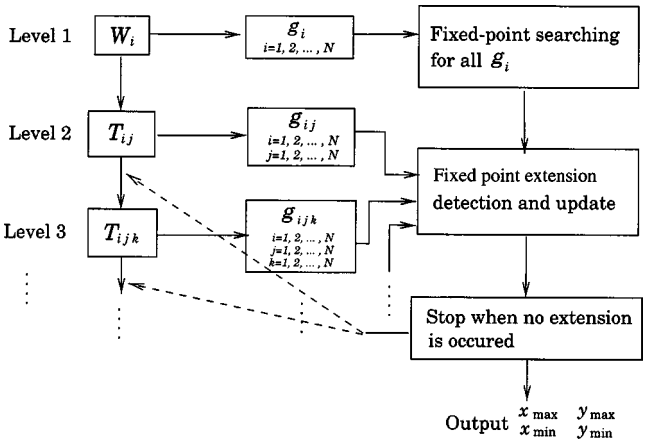

Fig. 4 Block diagram of the proposed HFPS method with detection of contour extension.

/* Pseudo code for hierarchical fixed-point seeking method*/
Calculate fixed points for each CAT;
Plot level-one contour;
do {
    For all of the transforms whose fixed points contribute the extension of contour;
    Calculate next-level transforms;
    Calculate the corresponding fixed points;
    Plot next-level contour;
    }
while (the contour is extended)
Output $x_{\min}$, $x_{\max}$, $y_{\min}$, *and* $y_{\max}$

We also summarize the procedures of the proposed HFPS method as follows:

Step 1: The fixed points (x_i, y_i) for all CATs W_i are calculated by Eq. (3). Then we plot the level-one contour based on the fixed points.
Step 2: Apply Pei's parallel algorithm to generate $T_{ij}^{(2)}$ for each CAT W_i.
Step 3: For each $T_{ij}^{(2)}$, determine the corresponding fixed points $g_{ij}^{(2)}$. Compare with the fixed points in the previous level; check if the contour is extended or not.
Step 4: If the contour is extended, eliminate the fixed point at the previous level and replace with the newly calculated fixed points; record the $T_{ij}^{(2)}$, so that we can employ Pei's parallel algorithm to generate the sublevel CAT $T_{ijk}^{(3)}$. Otherwise, stop extending the CAT at the next level, and prune this branch with index *ijk*.

Step 5: By steps similar to steps 2 to 4, perform the propagation of $T^{(n)}_{ijkl\ldots}$ and the fixed-point calculation recursively until no extension of the x and y ranges has occurred in the current step.
Step 6: The points in the final contour are used to obtain the maximum and the minimum values of the x and y coordinates, which specify the size and the coordinates of the fractal image.

3.2 *IFS Code Modification*

Once the coordinates $x_{\min}$ and $y_{\min}$ have been determined, we shift the corresponding point to the origin of coordinates in the original CAT. The method for transforming the coordinates in the CAT has been given in Ref. 1. We modify the parameters e and f in the IFS code as follows. Let (x',y') denote the original point (x,y) in the new coordinate system. That is, $(x',y')=\theta(x,y)$ denotes the new coordinate of the point. Let $W'(x',y')$ denote the same transformation as W but expressed in the new coordinate system. Then the relationship between the two CATs in the two coordinate systems is expressed by[1]

$$W'(x',y')=(\theta\circ W\circ\theta^{-1})(x',y'), \tag{7}$$

where the symbol $\circ$ denotes a transformation on the metric space. In our case, $\theta(x,y)=(x-x_{\min},y-y_{\min})$ and its inverse transform is $\theta^{-1}(x',y')=(x'+x_{\min},y'+y_{\min})$. Therefore, the new CAT W' becomes

$$\begin{aligned} W'\left(\begin{bmatrix}x'\\y'\end{bmatrix}\right)&=\mathbf{A}\begin{bmatrix}x'+x_{\min}\\y'+y_{\min}\end{bmatrix}+\mathbf{b}-\begin{bmatrix}x_{\min}\\y_{\min}\end{bmatrix}\\ &=\mathbf{A}\begin{bmatrix}x'\\y'\end{bmatrix}+(\mathbf{A}-\mathbf{I})\begin{bmatrix}x_{\min}\\y_{\min}\end{bmatrix}+\mathbf{b}=\mathbf{A}\begin{bmatrix}x'\\y'\end{bmatrix}+\mathbf{b}'. \end{aligned} \tag{8}$$

Obviously, the parameters in $\mathbf{A}$ are not changed, and the parameters in $\mathbf{b}$ should be modified as follows:

$$\begin{aligned} e'&=(a-1)x_{\min}+by_{\min}+e,\\ f'&=cx_{\min}+(d-1)y_{\min}+f. \end{aligned} \tag{9}$$

Suppose that the range of x and y determined from the HFPS method is $\{(x,y)|x_{\min}\leqslant x\leqslant x_{\max},y_{\min}\leqslant y\leqslant y_{\max}\}$. The original size becomes $P\times Q$, where $P=x_{\max}-x_{\min}$ and $Q=y_{\max}-y_{\min}$. To make all of the coordinates in the decoded image positive, we shift both $x_{\min}$ and $y_{\min}$ to zero. According to Eq. (9), the translation parameters e and f are modified to e' and f', respectively.

If the desired image is of size $M\times N$, then the magnification or contraction ratio becomes $R\times R$, where $R=\min\{M/P,N/Q\}$. The reason to choose the minimum of M/P and N/Q is that we have to magnify or contract the original image consistently in both the vertical and horizontal directions, to keep the shape of the scaled image the same as the original one. To fit the desired size, the new translation parameters e_n and f_n in the IFS code are calculated by

$$\begin{aligned} e_n&=e'R,\\ f_n&=f'R. \end{aligned} \tag{10}$$

The fractal image is then generated iteratively by the use of the modified IFS code $\{a,b,c,d,e_n,f_n\}$ without extra postprocessing.

3.3 *Computation Complexity*

We have denoted the iteration number as #itr. Suppose that we perform the shifting and scaling operation for a generated fractal image by the trivial algorithm. First of all, we have to find the coordinates of four corner points. The required computations for each generated point in the image include $8(\#\text{itr}-1)$ comparisons. Then, for each point, four multiplications and four additions are required in the scaling and translation operations defined in Eq. (1). Therefore, the total computation amount is $4\,\#\text{itr}$ multiplications and $4\,\#\text{itr}$ additions. Considering the proposed method, on the other hand, the computations for different fractal images depend on the following factors: (1) the number of determined fixed points, P_f, (2) the number L of levels searched in the HFPS method, and (3) the number of calculated sublevel CATs. To calculate a fixed point, as shown in Eq. (3) requires seven multiplications/divisions and seven additions/subtractions. Thus the total computations for the fixed points are $7P_f$ multiplications/divisions and $7P_f$ additions/subtractions. To check the extension of the contour, eight comparisons are made for each fixed point sought. Since the number of the fixed points sought is much smaller than the iteration number, i.e., $P_f\ll\#\text{itr}$, the comparison number in the proposed method is much smaller than that in the trivial method, especially for a large number of iterations. If the HFPS method stops at level L ($L\geqslant 2$), the number of calculated sublevel CATs is *at most** $N^{2^{L-1}}-N^{2^{L-2}}$, where N is the number of original CATs. As shown in Eq. (6), it requires 27 multiplications/divisions and 17 additions/subtractions to determine a sublevel CAT. Therefore, there are at most $27(N^{2^{L-1}}-N^{2^{L-2}})$ multiplications/divisions and $17(N^{2^{L-1}}-N^{2^{L-2}})$ additions/subtractions required in determining the sublevel CATs. The proposed method saves computation only when the total numbers of multiplications/divisions and of additions/subtractions are less than in the trivial method. That is,

$$\begin{aligned} &7P_f+27(N^{2^{L-1}}-N^{2^{L-2}})<4\,\#\text{itr}\\ &\qquad\text{for multiplications/divisions,}\\ &7P_f+17(N^{2^{L-1}}-N^{2^{L-2}})<4\,\#\text{itr}\\ &\qquad\text{for additions/subtractions.} \end{aligned} \tag{11}$$

We can expect extra saving on computations when the searched levels L are few.

*Only for the sublevel CATs whose fixed points contribute to the extension of the contour is calculation of their corresponding sublevel CATs required.

4 Quantization Effects

In the previous section, the IFS code and the coordinates of the generated image points were real numbers. However, real numbers must be quantized by a finite word length in hardware implementations. Therefore we investigate the quantization effects on the decoded fractal images. Suppose that a k-bit word length is used for the IFS code and the coordinates of the generated points. If we assume the image to be of size $M \times M$, the step size of quantization becomes $M/(2^k-1)$. The image resolution increases if we choose a higher k value. In Sec. 3, we found that the translation parameters e and f dominate the image scale. If the original size of the fractal image is small, e and f are also small, and vice versa. The quantization effects for large e and f will be more severe than those for small e and f. Since the fractal image is generated iteratively (point by point), the propagation problem of the quantization error occurs. If the word length for denoting the IFS code or the calculated x and y coordinates is not large enough, the quantization error of previously generated points will propagate to the current and following points. The decoded fractal image is composed of fewer points than in the ideal case. Therefore it is important to provide a large enough word length in hardware implementations. In our experiments, we examine the decoding results for different word lengths and determine what word length is enough to obtain a visually acceptable result.

Since the decoded image is binary, we also calculate the Hamming distance between the ideal image and the decoded images in the presence of quantization effects. Let the ideal fern image be within the range $x_{\min} \leqslant x \leqslant x_{\max}$ and $y_{\min} \leqslant y \leqslant y_{\max}$. In order to calculate the Hamming distance, we use the proposed scaling method to decode the image in the desired size $M \times M$. Since the size of the decoded image has to be fixed before the quantization, we only consider the error image with the same size as the ideal image. In a binary image, if a pixel is black, we assign it the value 0; if white, 1. The Hamming distance H between two binary images F_1 and F_2 is defined as

$$H = \sum_{0 \leqslant i,j < M} |F_1(i,j) - F_2(i,j)|, \tag{12}$$

where $F(i,j)$ denotes the (i,j)'th pixel value in the image F. In our experiments, we measure the normalized percentage error $\epsilon\%$ of the Hamming distance between the distorted and the ideal images, which is defined as

$$\epsilon\% = \frac{H}{M^2} \times 100\%. \tag{13}$$

5 Iteration Number in Decoding

The random iteration algorithm does not specify an iteration number. Therefore, it must have a convergence criterion to decide when the decoding process should stop. As shown in the section above, the generated points may overlap because of the quantization effects on the decoded image in practical implementations. If we choose a small iteration number, the decoded fractal image is sparse in its content. Otherwise, the decoding will be inefficient, since many computed points will overlap. In order to efficiently decode a fractal image, we should chose an adequate iteration number or find a convergence criterion for a given image size.

The convergence criterion has been clearly defined in the deterministic decoding algorithm as follow. Let the n'th iterated image be denoted as $F^{(n)}$. The error between the n'th and $(n-1)$'th decoded images can be represented by the Hamming distance defined in Eq. (12):

$$E(n) = \sum_{0 \leqslant i,j < M} |F^{(n)}(i,j) - F^{(n-1)}(i,j)|, \tag{14}$$

where $F^{(n)}(i,j)$ and $F^{(n-1)}(i,j)$ denote the (i,j)'th pixel values in the n'th and $(n-1)$'th decoded images, respectively. If the ratio

$$\gamma = \frac{|E(n) - E(n-1)|}{E(n-1)} \tag{15}$$

is less than a threshold value γ_{th}, the decoded image converges and the iteration process terminates. Otherwise, the iteration continues.

The random iteration algorithm does not specify a convergence criterion to terminate the iteration process. Therefore, we propose an efficient method to terminate the iteration process rather than set an iteration number according to the required resolution or our experience. In the random iteration algorithm, the reconstructed image is generated point by point. We count the overlap points during the iteration process. Suppose that the image size and resolution have been chosen. We first monitor each point and count the number of overlap points, $O(n)$, in the n'th iteration. The overlap points are the points repeatedly generated during the iteration process. Let $P(n)$ be the number of pixels exactly contributing to the fractal image. If the ratio

$$\eta(n) = \frac{O(n)}{P(n)} \tag{16}$$

is greater than a threshold η_{th}, the iteration stops. Otherwise, it continues.

The iteration numbers for different fractal images will depend on the desired scale and resolution of the fractal images. A sparse fractal image such as the snowflake may require only 3000 iterations. On the other hand, the castle requires more than 9000 iterations, since it is much denser. This difference can be inferred from their fractal codes. First of all, we calculate the absolute value of the determinant, $\rho_i = |ad - bc|$, of the matrix $\mathbf{A}$ in each CAT W_i. The physical meaning of ρ_i is the weight corresponding to the entire fractal image. These absolute values are then summarized as the contractivity sum ρ for comparison. That is,

$$\rho = \sum_{i=1}^{N} \rho_i. \tag{17}$$

Theoretically, a larger contractivity ρ should correspond to a larger area for the fractal image. However, some subimages may overlap when they are chosen in the encoding

Table 3 The calculated contractivity sums for four fractal images: Sierpiński triangle, fern, castle, and snowflake.

Image	ρ
Sierpiński triangle	0.75
Fern	0.93
Castle	0.91
Snowflake	0.55

stage.[1] Four contractivity sums corresponding to four sets of fractal codes are listed in Table 3. As we can expect, the required iteration number for the fern and castle images is more than for others, since those two images have larger contractivity sums. However, in the encoding stage of the fern image, the two subimages corresponding to W_1 and W_2 partially overlap[1] that for W_4. Therefore, the effective contractivity sum of the fern image should be smaller than that of the castle image. Thus the castle image should require the largest iteration number among the four fractal images.

6 Simulation Results and Discussion

The four IFS codes shown in Table 1 were used to test the proposed methods. First of all, the proposed HFPS method was employed to determine the original sizes and coordinates of fractal images from their IFS codes only. The contours determined for four fractal images are shown in Fig. 5. For different CATs, their fixed points are represented by different symbols. The level-one contour is plotted by a dotted line, and the level-two contour is plotted by a dashed line. In Figs. 5(a), 5(c), and 5(d), the extension of the contour stops at level one. That is, the fixed points of the level-two CATs do not contribute to the extension of the contour. As shown in Fig. 5(b), the extension of the contour for the fern image stops at level three. The reason is that, in the encoding stage, the fixed points are usually selected at the

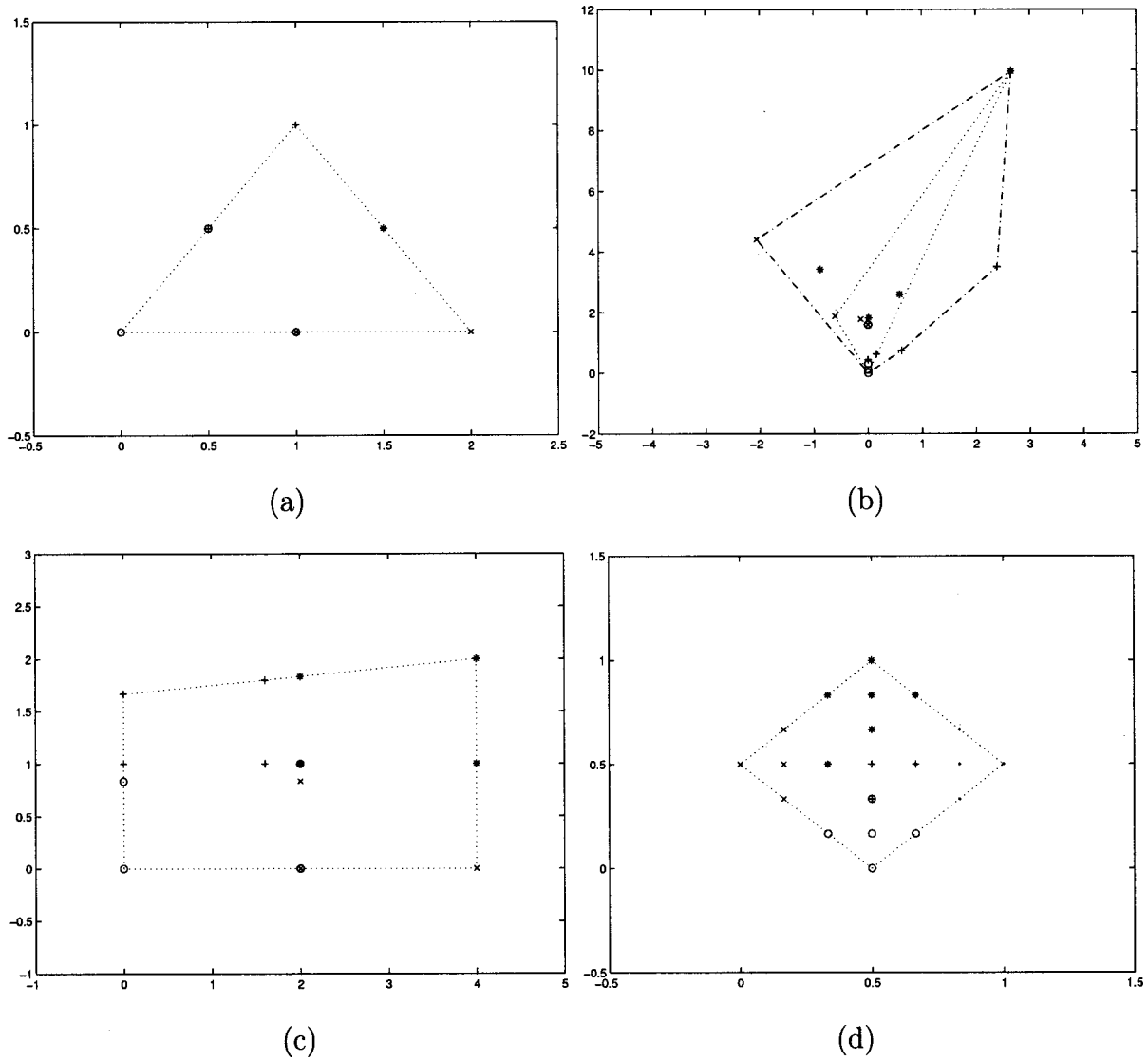

Fig. 5 The contours determined by the proposed HFPS method: (a) Sierpiński triangle, (b) fern, (c) castle, and (d) snowflake.

Table 4 The number of the fixed points sought for four fractal images.

Image	No of points sought				
	Level 1	Level 2	Level 3	Level 4	Total
Sierpiński triangle	3	9	0	0	12
Fern	4	16	8	0	28
Castle	4	26	0	0	20
Snowflake	5	25	0	0	30

corners of the image for easy encoding. Therefore, only a few levels are utilized in our HFPS method. Table 4 shows that only a few points are calculated on the different levels for the four fractal images. The proposed HFPS method is efficient because of its low computation load. The x and y ranges obtained from Fig. 5 are almost the same as those shown in Table 2, which are obtained from the decoded images.

We modify the translation parameters e and f to e' and f', and then to e_n and f_n. Table 5 shows the modified parameters in the corresponding CATs for the four fractal images. The generated fern image based on the modified coefficients e' and f' is shown in Fig. 6, in which we can see that the coordinates of the fern image are all positive. On the other hand, in Fig. 7 the decoded images are scaled to the same size, 512×512. Therefore, we can in fact rescale and relocate the fractal images by using the proposed HFPS method and the modification of IFS code.

Figure 8 shows the computations of the trivial methods and the proposed method under different numbers of CATs (N) and searched levels. Three iteration numbers are used in the trivial methods: 9000, 90,000, and 900,000. The

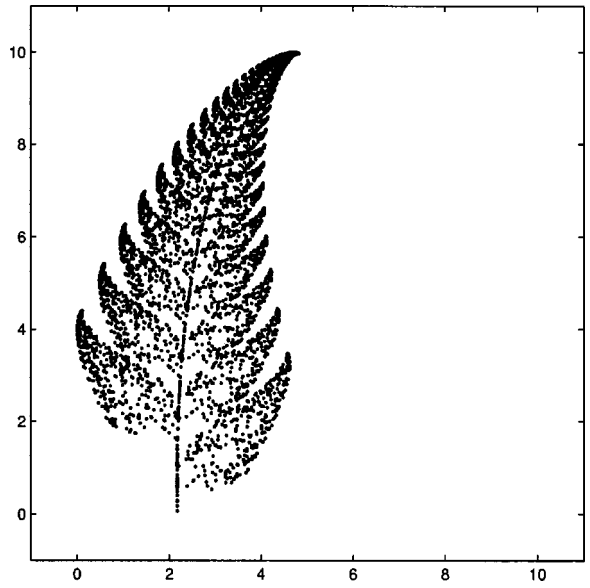

Fig. 6 The shifted fern image decoded with the modified IFS code $\{a,b,c,d,e',f'\}$.

number of computations for comparisons is about eight times the number of iterations. In addition, the multiplication/division and addition/subtraction numbers are proportional to the number of iterations and are the same. As shown in Table 4, the number of comparisons in the proposed method is at most $8\times30=240$, which is much less than $8\times9000=72{,}000$. On the other hand, the numbers of multiplications/divisions and additions/subtractions in the proposed method depend on the number of fixed points, the number of sublevel CATs, and the searched levels in

Table 5 The modified translation parameters in IFS code. Here e' and f' are calculated after the change of coordinates, and e_n and f_n are calculated for resizing the original fractal image. Note that the magnification/contraction ratio R is determined from the desired image size 512×512.

Image	W	e'	f'	R	e_n	f_n
Sierpiński triangle	1	0	0	256	0	0
	2	1	0		256	0
	3	0.5	0.5		128	128
Fern	1	2.178	0	51.2	139.162	0
	2	1.744	1.099		89.293	56.248
	3	2.507	−0.127		128.358	−6.492
	4	0.327	1.687		16.742	86.385
Castle	1	0	0	128	0	0
	2	2	0		256	0
	3	0	1		0	128
	4	2	1		256	128
Snowflake	1	0.333	0	512	170.496	0
	2	0	0.333		0	170.496
	3	0.333	0.333		170.496	170.496
	4	0.333	0.666		170.496	340.992
	5	0.666	0.333		340.992	170.496

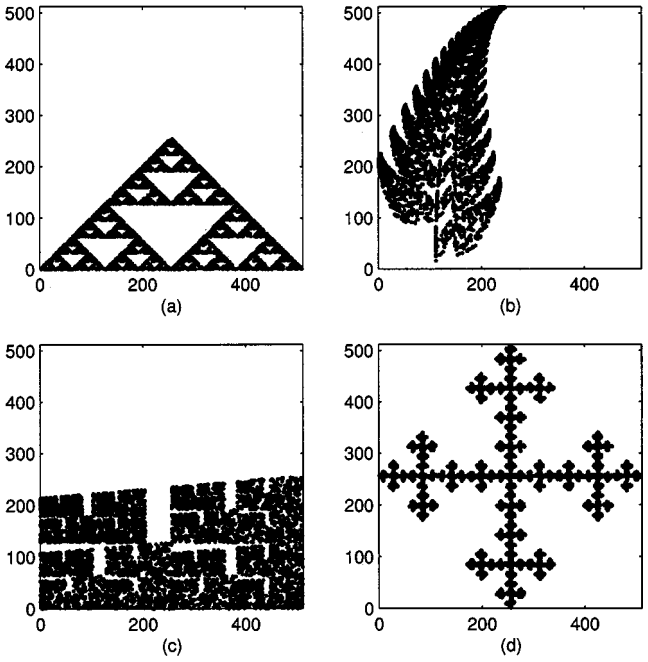

Fig. 7 The fractal images decoded by the use of the modified coefficients e_n and f_n shown in Table 5. The four images are of the desired size 512×512, and their shapes are identical to the original ones.

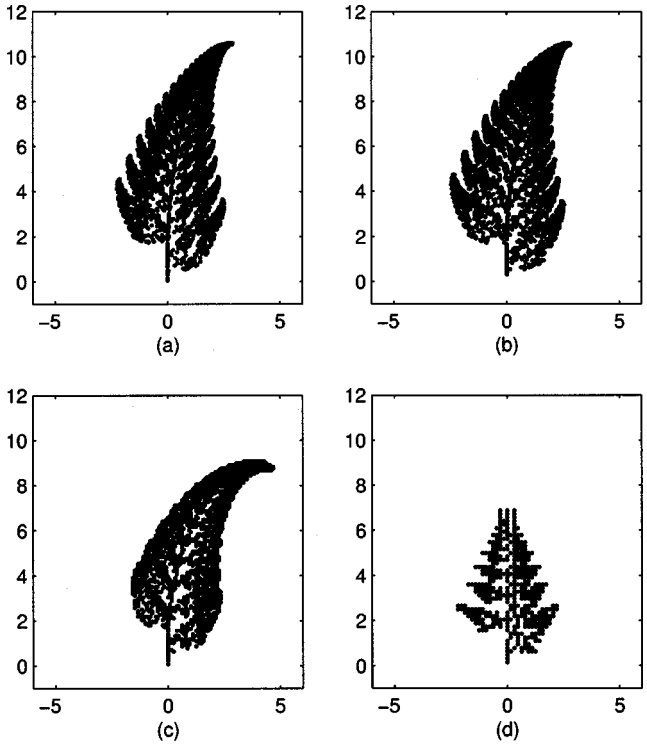

Fig. 9 The decoded fractal fern images corresponding to the quantization effects under different word lengths: (a) 9 bits, (b) 8 bits, (c) 7 bits, and (d) 6 bits.

the HFPS method. In the proposed HFPS method, only the sublevel CATs of the CATs related to the fixed points that contribute to the extension of the contour are calculated. Generally, only parts of the sublevel CATs are determined, so the actual number of computations should be much less than given by the curves in Fig. 8. Since the number of determined fixed points is much smaller than the iteration number, the number of comparisons in the proposed method is negligible and is not shown in the figure. The curves shown in fact are upper bounds on the numbers of computations for different cases. Furthermore, the searched levels in the HFPS method for most of the images are usually fewer than four. Therefore, our method is more efficient than trivial methods, especially when we need to generate high-resolution fractal images (where the iteration number will be much greater than 90,000).

The fern image is used to examine the quantization effects on the decoded results. Figures 9(a) to 9(d) show the decoding results for word lengths of 9, 8, 7, and 6 bits, respectively. It is clear that the decoded image quality is

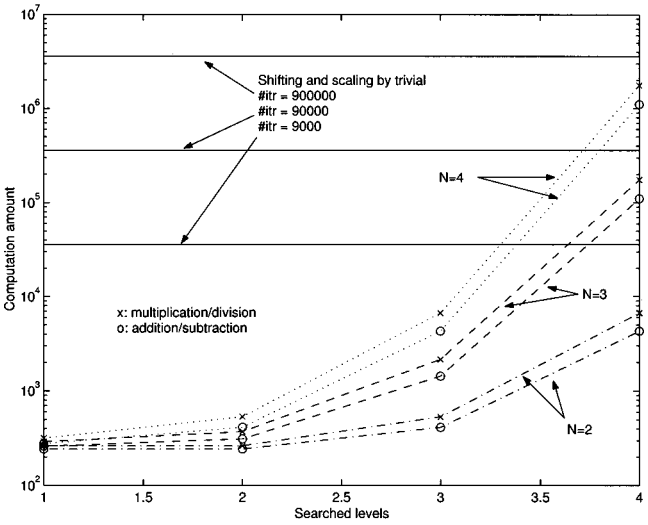

Fig. 8 Comparison of the computations in the trivial method and the proposed method under different numbers of CATs (N) and searched levels.

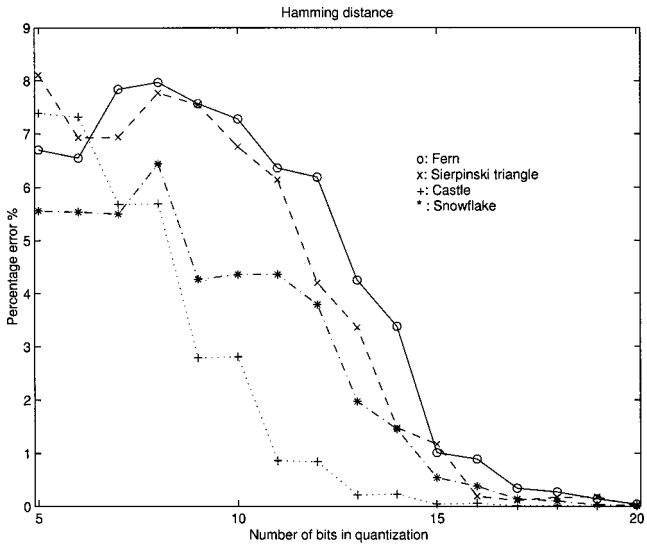

Fig. 10 The Hamming distance (presented as a percentage error) between the ideal and quantized images.

Table 6 Required iteration number at different threshold values η_{th} for various fractal images.

	Iteration number									
Image	$\eta_{th}=0.1$	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Sierpiński triangle	5056	10227	14650	19914	24498	28704	32532	37145	41244	45040
Fern	4536	11591	17690	24542	31709	37144	48789	55383	66033	75856
Castle	12650	36790	60592	88392	119417	NA	NA	NA	NA	NA
Snowflake	2326	4695	6538	8459	10517	12751	14252	16168	18751	20252

visually acceptable when the word length is 9 or 8. If the word length is less then 8, the quantization effects are severe and the image shape clearly changes. In Fig. 9(d) the quantization effects are especially severe, since the decoded points can only appear in some fixed positions.

Figure 10 gives a comparison of different word lengths. The reconstructed image size is 512×512, and the iteration number is set as 15,000. Figure 10 shows that to effectively eliminate the quantization error, a bit number more than 16 is required in practical implementation, so that the distortion can be neglected.

Considering the convergence criterion for the random iteration algorithm, we employ Eq. (16) to determine the iteration number under different threshold values $\eta_{th}=0.1$ to 1.0 for four test images. Table 6 shows the iteration number required for different threshold values η_{th} and different fractal images. For a given threshold value η_{th}, the iteration number depends on the image. The iteration number for the castle image becomes unavailable when the threshold value is greater than 0.5. Thus we refrain from measuring it and represent it by ''not available'' (NA). The simulation results are reasonable, since the contents and the sizes of different fractal image are also different. Thus the iteration number in conventional decoding algorithms is also image-dependent. For fractal images with a large contractivity sum (for example, the fern and castle), the required iteration number is also large. On the other hand, a small contractivity sum will correspond to a small iteration number. The simulation results thus verify our statements in Sec. 5.

7 Conclusion

In this paper we have dealt with the problem of displaying fractal images whose original sizes and coordinates are different. We propose a novel HFPS method to determine the original size and the coordinates of the fractal image from its IFS code. Two parameters of the IFS code, *e* and *f*, are then modified according to the original size and the desired size by the use of a coordinate transformation. Therefore the image can be decoded with a desired size and shown in a desired location. Comparisons of computations between the proposed method and the trivial methods are provided to verify the efficiency of our method.

We also investigate the quantization effects of the finite word length in hardware implementations. The percentage Hamming distance is used to measure the distortion between the ideal and quantized images. The simulation results suggest the required number of quantization bits for a real implementation, such that we can neglect the quantization error. Finally, we propose a convergence criterion that can automatically terminate the iteration process in the random iteration algorithm. A fractal image with a large contractivity sum requires a large iteration number, and vice versa.

The proposed HFPS method can be extended to three-dimensional fractals.

A fractal image can be seen as a single object in an image synthesis framework. We can integrate different fractal images into a single image frame. If their original sizes and coordinates are different, it is necessary to rescale and relocate the original fractal images so that we can put them together. In our future work, we will focus on performing arbitrary geometric transformations of fractal images such as rotation, skewing, reflection, and so on. These transformations of original fractal images can be performed by modifying the IFS code in advance without postprocessing the decoded results.

Acknowledgment

This research was partially supported by the National Science Council, Taiwan, under contract NSC 89-2213-E-324-028. Part of this work was presented at the 1999 IEEE International Conference on Image Processing.

References

1. M. Barnsley, *Fractals Everywhere*, 2nd ed., Academic Press, Boston (1993).
2. M. F. Barnsley and A. Sloan, ''A better way to compress images,'' *BYTE* **13**, 215–223 (1988).
3. M. Barnsley and L. Hurd, *Fractal Image Compression*, A. K. Peters, Wellesley, MA (1993).
4. M. Kawamata, H. Kanbada, and T. Higuchi, ''Determination of IFS codes using scale-space correlation functions,'' in *Proc. IEEE Workshop on Intelligent Signal Processing Communication Systems*, pp. 219–233, Taipei, Taiwan (1992).
5. S. Pei, C. Tseng, and C. Lin, ''Wavelet transform and scale space filtering of fractal images,'' *IEEE Trans. Image Process.* **4**(5), 682–687 (1995).
6. R. Rinaldo and A. Zakhor, ''Inverse and approximation problem for two-dimensional fractal sets,'' *IEEE Trans. Image Process.* **3**(6), 802–820 (1994).
7. H. A. Cohen, ''Deterministic scanning and hybrid algorithms for fast decoding of IFS encoded image sets,'' in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'92)*, Vol. III, pp. 509–512 (1992).
8. S. Pei, C. Tseng, and C. Lin, ''A parallel decoding algorithm for IFS codes without transient behavior,'' *IEEE Trans. Image Process.* **5**(3), 411–415 (1996).
9. H. T. Chang and C. J. Kuo, ''A fully parallel algorithm for fractal image decoding,'' in *1997 IEEE Proc. Int. Conf. on Circuits and Systems*, Vol. 2, pp. 1277–1280, Hong Kong (1997).
10. J. Tanida, A. Uemoto, and Y. Ichioka, ''Optical fractal synthesizer: concept and experimental verification,'' *Appl. Opt.* **32**(5), 653–658 (1993).
11. H. T. Chang and C. J. Kuo, ''An optical decoding architecture for the random iteration algorithm of iterated function system codes,'' *Opt. Rev.* **1**(2), 146–149 (1994).
12. H. T. Chang and C. J. Kuo, ''Random iteration algorithm based opti-

cal parallel architecture for fractal image decoding by use of IFS codes,'' *Appl. Opt.* **37**(8), 1310–1318 (1998).

Hsuan T. Chang received his BS degree in electronic engineering from National Taiwan Institute of Technology, Taiwan, in 1991, and MS and PhD degree in electrical engineering from National Chung Cheng University (NCCU), Taiwan, in 1993 and 1997, respectively. He was an adjunct lecture in Department of Automation in National Yun-Lin Polytech Institute from 1993 to 1994 and a visiting researcher at Laboratory for Excellence in Optical Data Processing, Department of Electrical and Computer Engineering, Carnegie Mellon University, from 1995 to 1996. Dr. Chang was an assistant professor in EE Department of Chein Kuo Institute of Technology (CKIT), Changhua Taiwan, from 1997 to 1999. He currently is an assistant professor in the Department of Information Management, Chaoyang University of Technology (CYUT), Wufeng Taichung, Taiwan. He is also an adjunct assistant professor at Graduate Institute of Communications Engineering of NCCU. Dr. Chang's interests include image/video processing, optical information processing/computing, medical image processing, and grey theory. He directed the Signal Processing Laboratory in the Mechatronic and Optical Technology Research Center at CKIT. He served as the reviewer of several international journals and was an invited speaker and program committee in local conferences. Dr. Chang is a member of SPIE, OSA, and the Chinese Image Processing and Pattern Recognition Society.