Paper:

Fractal Image Coding with Simulated Annealing Search

Shen Furao*, **and Osamu Hasegawa****

*Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology (TIT)
R2-52 Imaging Science and Engineering Lab., 4259 Nagatsuta, Midori-ku, Yokohama 226-8503, Japan
E-mail: furaoshen@isl.titech.ac.jp
**Imaging Science and Engineering Lab., Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama, Kanagawa 226-8503, Japan
[Received July 8, 2004; accepted November 30, 2004]

The main shortcomings of fractal image coders are (1) the slow speed for searching domain block pool, and (2) known fast algorithms leading to a loss of image quality. We propose efficient fractal image coding using simulated annealing method. Compared to previous schemes, our proposal greatly increases the search speed of domain block pool with almost no image quality loss. Experimental results indicate the high feasibility of the proposed method, which is, furthermore, extendable to other fractal coders.

Keywords: fractal image coding, simulated annealing, global optimal searching, iterated function system (IFS)

1. Introduction

Fractal image coding has been used in many image processing applications such as feature extraction [1], image signature [2], image retrieval [3, 4], and texture segmentation [5]. It has the advantages of very fast decompression, potentially very high compression ratios [6], and multiresolution property, with which an image can be decoded at higher or lower resolution than the original and can be zoomed in on sections of the image [7]. These advantages make fractal image coding very attractive in multimedia applications: Microsoft, for example, adopted it for compressing thousands of images in its *Encarta* multimedia encyclopedia [8].

Natural image features such as straight edges and constant regions are unchanged by rescaling and this redundancy is not exploited by transform coders. This local scale invariance is a major feature of fractal image coding. Although pure fractal coders have poorer performance than wavelet transform, hybrid fractal coders based on transform coding and fractal image compression significantly improve image quality, providing better compression than transform coders [9]. Such a coder proposed by Zhao and Yuan combined discrete cosine transform and block-based fractal coding [10]; Li and Kuo's fractal-wavelet coder [11] performed better than EZW [12].

Based on the idea of affine redundancy in real-world images, Barnsley and Sloan [13] originally proposed fractal image coding in 1988, attempting to set up an iterated function system (IFS) of one image to compress that image. In 1989, Jacquin [14] proposed a fractal compression scheme based on the block search that suited computer processing and established the bases of fractal image compression. He separated one image into nonoverlapping range blocks, then searched the domain block pool globally or partially to obtain optimal affine transformation.

The drawback of Jacquin's scheme was that block matching is very time-consuming. Much research has thus focused on how to speed up fractal image coding and most explore how to reduce the amount of domain blocks in the domain block pool. Some methods classify image blocks in some way. A range block was compared to domain blocks in the same class. Jacquin [15], for example, classified blocks on their edge content and Jacobs et al. [16] used block brightness orientation. Monro et al. [17] localized the domain pool with domain blocks closing to a given range block. Saupe [18] excluded domain blocks with the smallest variance. Tong and Pi [19] only searched domain blocks that satisfied specific adaptive search condition. Shen and Hasegawa [20] designed a nonsearch scheme with the help of the quadtree to speed up coding. The drawback of these schemes was that they dealt with only partial search and could not converge to the global optimal solution. The increase in search speed was also invariably accompanied by a loss of image quality.

Our target here was not to design a fractal image coder that attained a higher compression ratio or higher reconstruction quality but to propose a general technique that can be used for nearly all fractal coders to speed up the search of domain block pool with minimal loss of image reconstruction fidelity. We use simulated annealing, one of the best global search methods, to search the domain block pool globally. It greatly increases search speed while maintaining high image quality.

This paper is organized as follows: Section 2 analyzes the theoretic background of fractal image coding and simulated annealing. Section 3 defines a benchmark scheme compared to the proposed simulated annealing search algorithm. Section 4 discusses processing of a standard Lena image by the benchmark algorithm and simulated annealing algorithm. A comparison of these two schemes demonstrates the proposed scheme's good performance.

2. Theoretical Foundations

2.1. Fractal Image Coding

We assume that the size of original image f is $M \times M$. During fractal image coding, image f is segmented into nonoverlapping range blocks. Each range block is $B \times B$ in size. For each range block $f|R_k$, we search the domain block pool to get one domain block $f|D_k$, then apply contractive affine transformation W_k to $f|D_k$. The optimal selection of $f|D_k$ and W_k must ensure that image $g = \{W_k(f|D_k), k = 1, 2, \ldots, N\}$, which is obtained by transforming $f|D_k$ with $W_k(k = 1, 2, \ldots, N)$, has the minimum difference from $f = \{f|R_k, k = 1, 2, \ldots, N\}$. $f|D_k$ must be larger than $f|R_k$ to make W_k contractive. Usually we set the size of domain block as $2B \times 2B$. The domain block pool is extracted by sliding a window sized $2B \times 2B$ from the top left corner of original image f in integer step I_{step} horizontally or vertically. When I_{step} equals 1, the search is full and $(M - 2B + 1) \times (M - 2B + 1)$ domain blocks exist in the domain block pool.

If $W_k, k = 1, 2, \ldots, N$, is contractive, the set of all transformation $W = \{W_k, k = 1, 2, \ldots, N\}$ is also contractive. Based on the contractive mapping fixed-point theorem [21], there is one unique image f^* and $W(f^*) = f^*$. The collage theorem [13] guarantees that f^* will approximate original image f. Transformation W_k and the position of domain block $f|D_k$ constitute the fractal code of range block $f|R_k$. Fractal codes of all range blocks form the iterated function system (IFS) of image f.

Contractive affine transformation W_k consists of spatial contractive transform S_k, gray-level transform G_k, and geometric transform F_k.

Spatial contractive transform S_k maps $f|D_k$ (with size $2B \times 2B$) to D_k (with size $B \times B$). We define gray-level $g(i, j), (i = 1, 2, \ldots, B; j = 1, 2, \ldots, B)$ of D_k as the mean of four corresponding neighboring pixels of $f|D_k$, i.e.,

$$g(i,j) = \frac{1}{4} \sum_{(p,q)\in u_k^{-1}(i,j)} f(p,q) \quad \ldots\ldots\ldots \quad (1)$$

where u_k is mapping from $f|D_k$ to D_k.

Gray level transform G_k is defined by linear function $G_k(g(x,y)) = ag(x,y) + b$. We choose a and b to make

$$\int_{R_k} |f(x,y) - G_k(g(x,y))|^2 dL = \int_{R_k} [f(x,y) - ag(x,y) - b]^2 dL \quad \ldots\ldots \quad (2)$$

smallest.

Geometric transform F_k rotates or reflects D_k with 8 transformations: identity transform; rotation through 90°, 180° and −90°; reflection about mid-vertical axis, mid-horizontal axis, and two diagonal axies.

When determining the best matching domain block for one range block, we store W_k and the position of that best domain block as the fractal code of that range block. The basic fractal image coding algorithm is described as:

Algorithm 2.1: Basic fractal image coding algorithm

1. Separate image f into nonoverlapping range blocks $f|R_k$ and overlapping domain blocks $f|D_k$.
2. For every $f|R_k$, search the domain block pool to find the optimal selection of $f|D_k$ to minimize $error(W_k(f|D_k), f|R_k)$. Here, W_k is the contractive affine transformation and the error function is defined by

$$error(f,g) = \frac{1}{B^2} \sum_{i=1;j=1}^{B} (f(i,j) - g(i,j))^2. \quad (3)$$

3. Record the position of optimal domain block and W_k as the fractal code of $f|R_k$. Then go to step (2) to process the next range block $f|R_{k+1}$ until all range blocks are processed.

2.2. Simulated Annealing

Simulated annealing (SA), a Monte-Carlo method, was developed to statistically find the best global fit of a nonlinear nonconvex cost-function over a d-dimensional space [22]. We assume that Ω is one area of the d-dimensional space and $x \in \Omega$. Look for x_{min} to make $f(x_{min}) = \min f(x)$, where $f(x)$ is the cost-function and has many local optimal solutions. Simulated annealing starts at one arbitrary $x_0 \in \Omega$, then randomly proceeds following the statistical property of cost-function. This algorithm enables an annealing schedule for "temperature" T decreasing slowly to avoid the local optimal point. Simulated annealing consists of the following three functional relationships:

- $g(x)$: Generation function. Probability density of state-space of d parameters $x = x^i; i = 1, \ldots, d$. This function is used to generate a new state of x.
- $h(x)$: Acceptance function. Probability for acceptance of new cost-function given the immediately previous value. This function determines if the new state is accepted and can take place of the previous state.
- $T(k)$: Annealing schedule. Schedule for "annealing" "temperature" T in annealing-time steps k, i.e., changing the volatility or fluctuations of one or both of the two previous probability densities.

3. Simulated Annealing Search for Fractal Image Coding

Here, we define one benchmark fractal image coding scheme, then give an algorithm for how to use simulated annealing to search the domain block pool.

3.1. Benchmark Scheme

In range blocks and contractive domain blocks, there are $n = B \times B$ pixel intensities, $r_1, \ldots, r_n$ (from R_k) and $d_1, \ldots, d_n$ (from D_k). In Algorithm 2.1, the error function is written as:

$$error = \frac{1}{n}\sum_{i=1}^{n}(ad_i + b - r_i)^2. \quad \ldots\ldots\ldots (4)$$

The Least Squares method is used to minimize eq.(4), with the result that:

$$a = \frac{n\sum_{i=1}^{n} d_i r_i - \sum_{i=1}^{n} d_i \sum_{i=1}^{n} r_i}{n\sum_{i=1}^{n} d_i^2 - (\sum_{i=1}^{n} d_i)^2} \quad \ldots\ldots (5)$$

$$b = \frac{\sum_{i=1}^{n} r_i - a\sum_{i=1}^{n} d_i}{n} \quad \ldots\ldots\ldots\ldots (6)$$

From least squares results, we find a high correlation between a and b. During the quantization of a and b, for a given total number of bits, allotting more bits for one parameter will influence the other parameter. To eliminate the correlation between a and b, we adopt the following gray level transformation proposed by [23] and advocated by [19, 20]:

$$f(D) = a(D - \bar{d}I) + \bar{r}I. \quad \ldots\ldots\ldots\ldots (7)$$

Here, D is the contractive domain block, $\bar{r}$ is the range block mean, $\bar{d}$ is the contractive domain block mean, and I is a block whose elements are all ones. With this gray level transform, matching error is defined by

$$E = ||a(D - \bar{d}I) - (R - \bar{r}I)||^2. \quad \ldots\ldots\ldots (8)$$

Tong and Pi [19] conducted experiments to prove that there is nearly no correlation between a and $\bar{r}$ in eq.(7).

Based on the above analysis, we define the following scheme as the benchmark scheme:

Algorithm 3.1: Benchmark Scheme

1. Separate the original image into range blocks and domain blocks.
2. For each range block, encode its block mean $\bar{r}$.
3. Fully search the domain block pool to find the best matching domain block minimizing eq.(8).
4. Encode the position of the best matching domain block and associated scaling parameter a, then go to step (2) to encode the next range block.

3.2. Simulated Annealing Scheme

From Algorithm 2.1 and Algorithm 3.1 we find that, during encoding, for every range block, we must search for the best matching domain block in the domain block pool. The domain block pool consists of many domain blocks, meaning that fractal image coding is a global optimal problem: for every range block, find the global optimal solution of the domain block to minimize the target function (8).

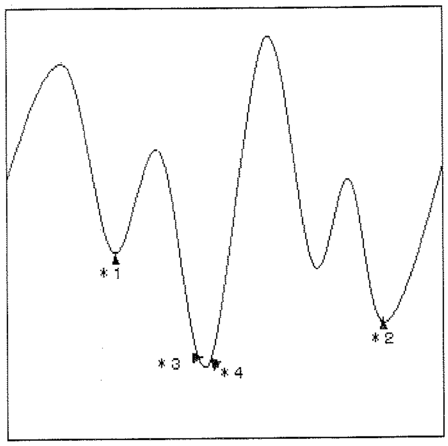


Fig. 1. SA versus local optimal methods: Local optimal methods are trapped in local minimum *1 or *2; SA escapes local solutions and converges to near global optimal minimum *3 or *4.

Many techniques are used to solve optimal problems. Well-known ones such as the Gauss-Newton algorithm and Levenberg-Marquardt Method involve local optimums trapping them in local minima. For Genetic Algorithm (GA) [24], the premature convergence of GA also leads to local minima.

Simulated annealing presents an optimization that: (a) processes cost functions possessing quite arbitrary degrees of nonlinearity, discontinuity, and stochasticity; (b) processes quite arbitrary boundary conditions and constraints imposed on these cost functions; (c) is implemented quite easily with a degree of coding quite minimal compared to other nonlinear optimization algorithms; and (d) statistically guarantees finding an optimal solution. **Fig.1** compares simulated annealing with other local optimal methods. Local optimal methods are easy trapped in local minima (such as *1, *2 in **Fig.1**), and simulated annealing escapes local solutions and converges to the global solution (such as *3, *4 in **Fig.1**). Standard simulated annealing is also useful in finding optimal block and row-column designs [25]. Based on the above analysis, we suggest that simulated annealing, one of the best global optimal search method, be used for this fractal encoding problem.

Based on the benchmark algorithm (Algorithm 3.1), we define eq.(8), a nonlinear nonconvex function over 2-dimensional space, as the cost-function of simulated annealing. For a given range block, we search the domain block pool to minimize the energy defined in (8). $x = \{x_1, x_2\}$ is defined by the position of the domain block. x_1 stands for the horizontal position, x_2 stands for the vertical position, and they are all limited in $[0, M - 2B]$. In practice, the following schemes are used to give generation function $g(x)$, acceptance function $h(x)$, and annealing schedule $T(k)$:

- Generation function $g(x)$: Based on the functional form derived for many physical systems belonging to the class of Gaussian-Markovian systems, the algorithm chooses for g,

$$g(x) = (2\pi T)^{-d/2} \exp[-\Delta x^2/(2T)] \quad . \ . \ . \quad (9)$$

where Δx is the deviation of x_{k+1} from x_k, and T is a measure of fluctuations of Boltzmann distribution g in the d-dimensional x-space. Because the computer generates only pseudorandom numbers, in practice, the following scheme is used to generate a Gaussian distribution from a uniform distribution. Assuming that η_1, η_2 are random data in $(0,1)$, conduct the transform

$$\xi_1 = \sqrt{T}(-2\ln\eta_1)^{1/2}\cos 2\pi\eta_2 \quad . \ . \ . \ . \ (10)$$

or

$$\xi_2 = \sqrt{T}(-2\ln\eta_1)^{1/2}\sin 2\pi\eta_2 \quad . \ . \ . \ . \ (11)$$

and new state x will be:

$$x_{k+1} = x_k + \xi_1(x_{max} - x_{min}) \quad . \ . \ . \ . \ . \ . \ (12)$$

or

$$x_{k+1} = x_k + \xi_2(x_{max} - x_{min}) \quad . \ . \ . \ . \ . \ . \ (13)$$

here, $x_{max,1} = x_{max,2} = M - 2B$, $x_{min,1} = x_{min,2} = 0$.

- Acceptance function $h(x)$: Based on the previous state E_k, the chance of obtaining a new state E_{k+1} is acceptance probability. h is defined by

$$h(x) = \frac{\exp(-E_{k+1}/T)}{\exp(-E_{k+1}/T) + \exp(-E_k/T)} = \frac{1}{1+\exp(\Delta E/T)} \quad . \ . \ . \ . \ . \ . \quad (14)$$

where ΔE represents the "energy" difference between the present and previous values of the cost-function appropriate to the physical problem, i.e., $\Delta E = E_{k+1} - E_k$. This is essentially the Boltzmann distribution contributing to the statistical mechanical partition function of the system.

In practice, we use

$$\exp(-\Delta E/T) \quad . \ . \ . \ . \ . \ . \ . \ . \ . \ . \ . \ (15)$$

to judge whether the new state will be accepted. When $\Delta E < 0$ or $\Delta E = 0$, the new state is accepted. If $\Delta E > 0$, assume ε is random data between $(0,1)$, if $\exp(-\Delta E/T) > \varepsilon$, the new state is accepted. This metropolis judgment rule ensures that the cost-function escapes local solutions and converges to the global optimal solution.

- Annealing schedule $T(k)$: It was proved [26] that the annealing schedule defined by

$$T(k) = T_0/\ln(k) \quad . \ . \ . \ . \ . \ . \ . \ . \ . \ . \ (16)$$

where T_0 is the initial temperature, ensures that, during annealing, each state of Ω has the possibility of being found. In practice, we use

$$T(k) = T_0/\ln(1+k) \quad . \ . \ . \ . \ . \ . \ . \ . \ (17)$$

to avoid overflow caused by $k = 1$. This change does not degrade the validity of (16).

We give the algorithm of using simulated annealing to search the domain block pool as follows:

Algorithm 3.2: Proposed scheme

1. Define the cost-function as

$$f(x) = \min_a ||a(D - \bar{d}I) - (R - \bar{r}I)||^2 \quad . \ . \ (18)$$

$x = \{x_1, x_2\}$ is the position of the domain block in original image f.

2. Initialize x, temperature T_0, maximum iterated times k_f of every temperature T.
3. For every range block, calculate its mean $\bar{r}$.
4. Annealing with annealing schedule (17).
5. Generate new states with generation function defined by eqs.(10), (12) or (11), (13).
6. Use Metropolis judgment rule formula (15) to judge whether the new state will be accepted. Then go to step (5) to do k_f searches.
7. If the cost-function is satisfied, store $\bar{r}$, a, and x; go to step (3) to process the next range block. If the cost-function is not satisfied, go to step (4) to continue annealing.

4. Experiment

4.1. Use Proposed Scheme for Full Search

In this experiment, we set the search space of the proposed scheme as the domain block pool of the benchmark full search fractal coder, then compare the benchmark full search scheme with the proposed search scheme for standard Lena ($256 \times 256 \times 8$).

In experiment, 2 bytes are used to store the position of the domain block (one for x_1 and one for x_2). Six bits are allocated to store the mean of the range block, and 2 bits to store gray-level transformation scaling parameter a. Thus there are only four values of a: 0.25, 0.5, 0.75, and 1.0. We choose *bpp* (bit per pixel) and PSNR (peak-to-peak signal-to-noise ratio) as the criteria of comparison. PSNR is defined as:

$$PSNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{N}\sum_{i=1}^{N}(f_i - f_i^*)^2}\right) \quad . . . (19)$$

where f_i is the pixel of the original image and f_i^* is the pixel of the decompressed image. N denotes the total number of pixels.

For $B = 4$, 8, and 16, we use benchmark full search Algorithm 3.1 and proposed search Algorithm 3.2 to process the standard Lena Image. To simplify the algorithm, for both algorithms, we use only the identity transform as the geometric transform.

For a benchmark full search, we conduct $(M - 2B + 1) \times (M - 2B + 1)$ searches to find the global optimal solution for every range block. In this experiment, $M = 256$. When $B = 4$, the benchmark scheme must search 62,001 times for every range block; when $B = 8$, it must search 58,081 times; when $B = 16$, it must search 50,625 times.

The statistical search of the proposed scheme saves search times while attaining nearly the same reconstruction quality of the benchmark scheme. Some aspects of the proposed scheme are very important:

- The initial value of position x influences convergence speed. We choose the top-left of the range block as the initial value of x, which is better than an arbitrary initial value.
- If the initial value of T_0 is large enough, results are satisfied. We use 1000, 3000, and 5000 as initial temperature T_0 and get nearly the same results, so we only report results with $T_0 = 3000$ here.
- The maximum iterated times k_f of one temperature influence the balance between encoding speed and image quality. If k_f is too small, cost-function cannot crystallize as the best state of that temperature. If k_f is too large, it slows the search significantly. We set $k_f = 100$ in this experiment. If we want better reconstructed image quality, we can set k_f larger but this slows encoding.
- During the experiment, the generation of new state and Metropolis Judgment rule depends on random data. We apply the mean of 5 random data to reduce the influence of computer pseudorandom numbers.

During the calculation of the spatial contractive transformation of domain block, some calculation is repeated. To eliminate repeat calculation before encoding range blocks, we preprocess original image f (with size $M \times N$) to image f_p with the same size of f to express every pixel $f_p(i, j)$ as the mean of the neighboring four pixels in the corresponding position of f [20]. For $i = 1, 2, \ldots, M; j = 1, 2, \ldots, N$

$$f_p(i, j) = \frac{1}{4}\sum_{m=0}^{1}\sum_{n=0}^{1} f(i+m, j+n). \quad . . . . . (20)$$

During range block encoding, assuming the position of domain block is (row_D, col_D), we need only to extract the contractive domain block from f_p as follows:

$$D(i, j) = f(row_D + 2i, col_D + 2j),$$
$$i = 1, 2, \ldots, B; \quad j = 1, 2, \ldots, B. \quad . . . . (21)$$

This eliminates repeat calculation in both the benchmark scheme and the simulated annealing search scheme.

In the experiment, software environment, the operating system is Windows 2000 and the programming language is Visual C++ 6.0. Hardware uses a PC with an Intel Pentium III 600MHz CPU and 128MB of memory. During the experiment, a full search is done for the benchmark scheme. For the proposed search scheme, we use different search times to test it. The results for Lena ($256 \times 256 \times 8$) are listed in **Tables 1-3** and **Figs.2-4**, where "FS" means "Benchmark Full Search" and, "SA" means "Simulated Annealing Search."

Table 1. Results for Lena ($256 \times 256 \times 8$), $B = 4$, $0.78(b/p)$ bit rate.

	Searches	PSNR	Encode seconds	Speedup ratio
FS	62001	33.639	1672	
SA	100	29.042	4	418
	500	30.493	22	76
	1000	31.178	46	36.35
	5000	32.549	228	7.33
	20000	33.616	972	1.72

Table 2. Results for Lena ($256 \times 256 \times 8$), $B = 8$, $0.329(b/p)$ bit rate.

	Searches	PSNR	Encode seconds	Speed up ratio
FS	58081	27.688	1186	
SA	100	24.446	3	395.33
	500	26.064	12	98.83
	1000	26.375	24	49.42
	5000	27.044	122	9.72

Table 3. Results for Lena ($256 \times 256 \times 8$), $B = 16$, $0.063(b/p)$ bit rate.

	Searches	PSNR	Encode seconds	Speed up ratio
FS	50625	23.185	973	
SA	100	21.509	2	486.5
	500	22.399	10	97.3
	1000	22.474	22	44.23
	5000	22.710	104	9.36

Table 1 shows that, for $B = 4$, even with only 100 searches for every range block, simulated annealing gets a reasonable PSNR and speed increases 418 times. The PSNR increases along with the increase in searches. In 5,000 searches, speed improves 7.33 times with only a slight loss of PSNR from the benchmark full search. In 20,000 searches, speed improves 1.72 times and attains nearly the same PSNR as the full search algorithm.

Comparing **Fig.2**, **Fig.3**, and **Fig.4**, we find that, the full search yields a good decoding image (**Fig.3**); With 20,000 searches, the proposed search attains nearly the

Fig. 2. Original Lena image ($256 \times 256 \times 8$).

Fig. 4. Decoding image of proposed method ($B = 4$, iterated 20000 times, searched 972 seconds, PSNR=33.616).

Fig. 3. Decoding image of benchmark full search ($B = 4$, searched 1672 seconds, PSNR=33.639).

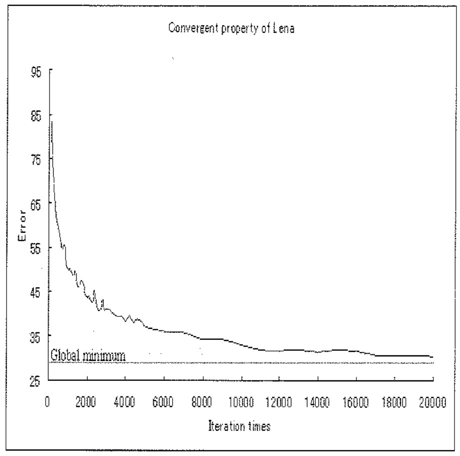


Fig. 5. Lena ($256 \times 256 \times 8$) convergence.

same results as the full search (**Fig.4**).

The results for $B = 8$ and $B = 16$ (in **Table 2** and **Table 3**) also prove that simulated annealing provides an effective global search for the block matching problem.

Figure 5 shows the convergence of the proposed search scheme. Error (defined in formula (3), measurement of decoding fidelity, low error means high decoding quality) decreases following the increase in the number of searches. The decreasing degree of error is smaller and smaller, and after a period of iteration, error converges to near a global optimal solution. The fluctuation in **Fig.5** means that, during the process, the proposed algorithm escapes the local solution frequently and converges to the global optimal solution.

To show that if the proposed scheme works for other images, with the same parameter settings of initial x, T_0, and k_f, we tested two other well-known images, i.e., Boat ($512 \times 512 \times 8$) (**Fig.6**) and Baboon ($512 \times 512 \times 8$) (**Fig.8**), with the proposed search algorithm and the benchmark full search algorithm. Results show that the proposed scheme attains nearly the same reconstruction quality as the full search scheme in only half search time. For reference, we give the decoding image of the proposed scheme for Boat in **Fig.7** and Baboon in **Fig.9**. **Fig.10** shows the convergence of Boat ($512 \times 512 \times 8$). **Fig.11** shows the convergence of Baboon ($512 \times 512 \times 8$). **Fig.10** and **Fig.11** also show that the proposed scheme escapes local solutions and converges to the global optimal solution, proving that simulated annealing suits the fractal image coding problem.

4.2. Use Proposed Scheme for Other Fractal Coders

In Section 4.1, we use the proposed search scheme for the benchmark full search and speed up the full search greatly with nearly no loss of image quality. In this section, we use the proposed search scheme for some other fractal coders to show how the proposed scheme speeds up fractal coders that must search a domain block pool with almost no loss of *bpp* and PSNR. We use Lena ($256 \times 256 \times 8$) as the test image.

- Jacob's local search scheme [16] skips adjacent blocks to decrease the number of domain blocks in

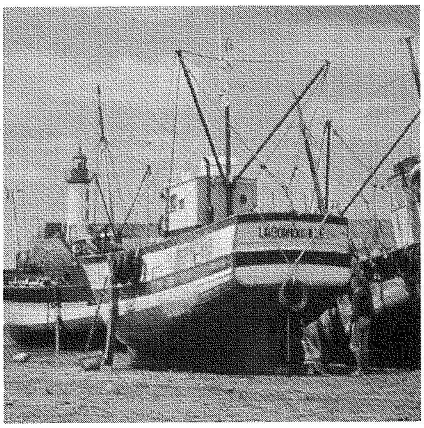


Fig. 6. Original boat image (512 × 512 × 8).

Fig. 7. Decoding of proposed scheme for **Fig.6** ($B = 4$, PSNR = 33.82).

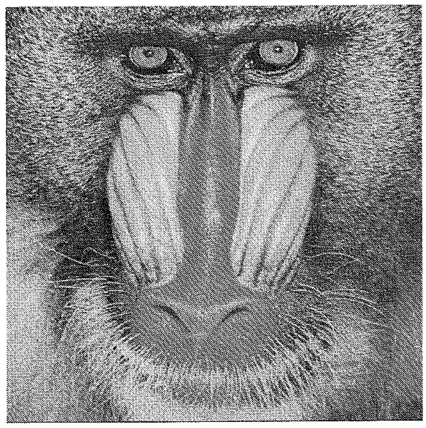


Fig. 8. Original baboon image (512 × 512 × 8).

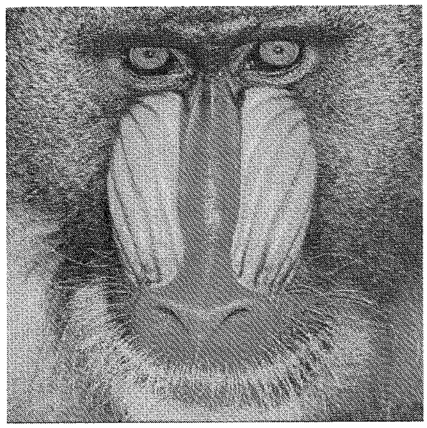


Fig. 9. Decoding of proposed scheme for **Fig.8** ($B = 4$, PSNR = 25.86).

the domain block pool and gets a subset of domain block pool, then full searches the subset. We realized Jacob's local search and compared the results with those for the proposed scheme in **Fig.12**. **Fig.12** shows that, to get the same PSNR, the proposed search only needs nearly $1/3$ the search time of the Jacob scheme.

- Tong and Pi [19] eliminate some domain blocks in the domain block pool by using adaptive conditions, use some preprocessing technique, and combine the adaptive search with Jacob's local search to speed up coding. Here, we add adaptive conditions to the benchmark full search scheme (without preprocessing and Jacob's scheme) and compare Tong and Pi's adaptive search with proposed simulated annealing search results in **Fig.13**. To get the same PSNR, the proposed scheme is much faster than adaptive search.

5. Conclusions and Discussion

The main problem of fractal image coding is slow matching for range and domain blocks. We proposed a search based on simulated annealing, which is one of the best global optimal methods, to search for the best matching domain block. We designed cost-function and 2-dimensional searching space to make the fractal encoding problem suitable for simulated annealing and conducted experiment to prove that simulated annealing suits fractal encoding problems. The proposed method improves encoding speed greatly with nearly no loss of image quality, and escapes local solutions and converges to the global optimal solution. The proposed scheme is also very easy to implement.

Our purpose was not to develop a new fractal coder with a higher compresion ratio or higher reconstruction quality, but to speed up the search of the domain block pool without sacrificing any image reconstruction fidelity. Thus, we do not use techniques such as quadtree to improve the performance of the proposed algorithm. It is very easy to replant the proposed scheme for nearly all fractal coders that must search a domain block pool.

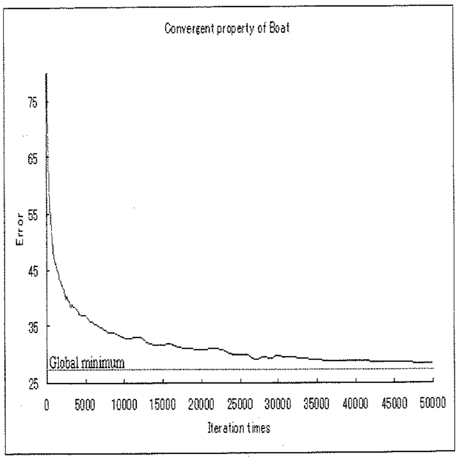


Fig. 10. Boat ($512 \times 512 \times 8$) convergence.

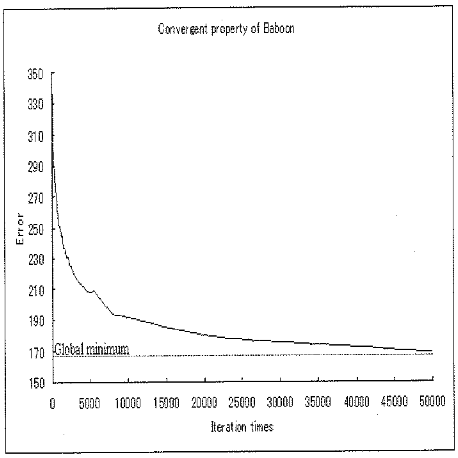


Fig. 11. Baboon ($512 \times 512 \times 8$) convergence.

Fisher's quadtree scheme [21], for example, must search a domain block pool for every uncovered range block, and using the proposed scheme is able to speed up the quadtree scheme greatly.

The proposed scheme can be used with other fast schemes. For example, after the subset of a domain block pool is determined, the proposed algorithm can be used to search the subset; adaptive conditions can also be added to as needed. It is probable that simulated annealing only converges to the near neighbor of the global optimal solution (such as *3, *4 in **Fig.1**). If we want a precise global optimal solution, SA will spend plenty of search time. We can accelerate the search by incorporating some local optimal methods to the simulated annealing, i.e., when simulated annealing converges to a near neighbor of the global solution, we begin the local search from the simulated annealing results. This is our next projected research topic.

Much research has been done on simulated annealing such as fast simulated annealing [27] and very fast simulated annealing [28], seeking to speed up the convergence of simulated annealing. Basu and Frazer [29] found the critical temperature and did considerable search around

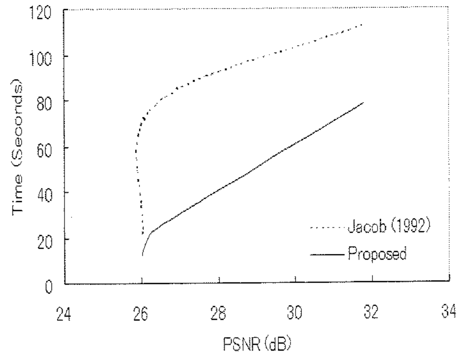


Fig. 12. Comparison of Jacob scheme with proposed search for Lena ($256 \times 256 \times 8$), $B = 4$, $bpp = 0.78b/p$.

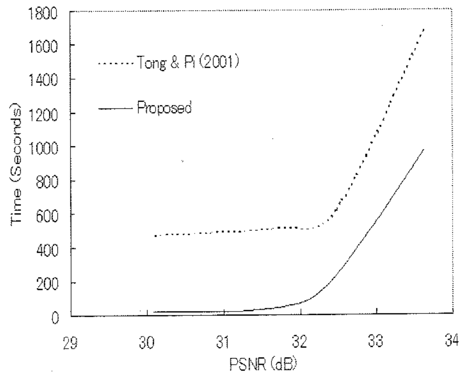


Fig. 13. Comparison of Tong and Pi scheme with proposed search for Lena ($256 \times 256 \times 8$), $B = 4$, $bpp = 0.78b/p$.

the critical temperature to improve search quality. This article has only realized classical Boltzmann simulated annealing for fractal image coding. In the special area of fractal image coding, it is worth researching how to converge simulated annealing quickly and how to improve encoding quality. Our work thus provides an initial exploration in this fascinating area.

References:

[1] B. Schouten, and P. de Zeeuw, "Feature extraction using fractal codes," in Visual 99; visual information and information systems, Third International Conference, Springer, Berlin, pp. 483-492, 1999.

[2] J. Puate, and F. Jordan, "Using fractal compression scheme to embed a digital signature into a image," Proc. SPIE, Vol.2915, pp. 108-118, 1997.

[3] M. Ancis, W. Buchwald, P. Giusto, D. Daniele, and D. Schmidt, "Fractal zooming of thumbnails for progressive image coding," in C.-C.J. Kuo, S.-F. Chang, S. Panchanathan (Eds.), Multimedia Storage and Archiving Systems III, Proceesings of the SPIE, Rhode Island, USA, Vol.3527, pp. 541-549, 1998.

[4] H.A. Cohen, "Thumbnail-based image coding utilizing the fractal transform," Proceedings of the ICIP-96 IEEE International Conference on Image Processing, Lausanne, September 1996.

[5] L.M. Kaplan, and C.C. Jay Kuo, "Texture segmentation via haar fractal feature estimation," J. Visual Commun. Image Representation, 6 (4), pp. 387-400, 1995.

[6] C.S. Tong, and M. Pi, "Analysis of a hybrid fractal-predictive-coding compression scheme," Signal Processing: Image Communication, 18, pp. 483-495, 2003.

[7] D.M. Monro, and P.D. Wakefield, "Zooming with implicit fractals," Proceeding of the ICIP-97 IEEE International Conference on Image Processing, Washington, DC, October 1997.

[8] M. Barnsley, "Fractal image compression," in J.M. Blackledge (Eds.), Image Processing: Mathematical Methods and Applications, Clarendon Press, Oxford, pp. 183-210, 1997.

[9] B. Wohlberg, and G.d. Jager, "A review of the fractal image coding literature," IEEE Trans. on Image Processing, Vol.8, No.12, pp. 1716-1729, 1999.

[10] Y. Zhao, and B. Yuan, "A hybrid image compression scheme combining block-based fractal coding and DCT," Signal Processing: Image Communication, 8, pp. 73-78, 1996.

[11] J. Li, and C.-C. Kuo, "Fractal wavelet coding using a rate-distortion constraint," Proc. IEEE Int. Conf. Image Processing, Lausanne, Switzerland, Vol.2, pp. 81-84, 1996.

[12] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Trans. on Signal Processing, Vol.41, No.12, pp. 3445-3462, 1993.

[13] M.Barnsley, and A. Sloan, "A better way to compress images," BYTE, pp. 215-223, Jan. 1988.

[14] A.E. Jacquin, "Fractal image coding: a review," Proc. of the IEEE, Vol.81, No.10, pp. 1451-1465, 1993.

[15] A.E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," IEEE Trans. Image Processing, Vol.1, No.1, pp. 18-30, 1992.

[16] E.W. Jacob, Y. Fisher, and R.D. Boss, "Image compression: a study of the iterated transform method," Signal Processing, Vol.29, pp. 251-263, 1992.

[17] D.M. Monro, and F. Dudbridge, "Approximation of image blocks," in Proc. Int. Conf. Acoustics, Speech, Signal Processing, Vol.3, pp. 4585-4588, 1992.

[18] D. Saupe, "Lena domain pools for fractal image compression," in Conf. Proc. SPIE Electronic Imaging'96, Still Image Compression II, Vol.2669, San Jose, CA, 1996.

[19] C.S. Tong, and M. Pi, "Fast fractal image encoding based on adaptive search," IEEE Trans. on Image Processing, Vol.10, No.9, pp. 1269-1277, 2001.

[20] S. Furao, and O. Hasegawa, "A fast no search fractal image coding method," Signal Processing: Image Communication, to appear.

[21] Y. Fisher, "Fractal image compression," SIGGRAPH'92 Course Notes, Vol.12, July 1-19, 1992.

[22] S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi, "Optimization by simulated annealing," Science, 220, (4568), pp. 671-680, 1983.

[23] G.E. Øien, and S. Lepsøy, "A class of fractal image coders with fast decoder convergence," in Y. Fisher (Eds.), Fractal Image Compression: Theory and Applications, New York: Springer-Verlag, pp. 153-174, 1995.

[24] J.D. Schaffer, L.J. Eshelman, and D. Offutt, "Spurious correlations and premature convergence in genetic algorithms," in G. Rawlins (Eds.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, pp. 102-112, 1991.

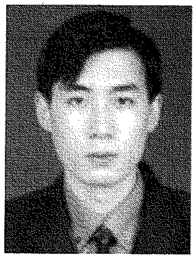
[25] W.N. Venables, and J.A. Eccleston, "Randomized search strategies for finding near optimal block and row-column designs," Report, University of Adelaide, Australia, 1992.

[26] S. Geman, and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images," IEEE Trans. Patt. Anal. Mac. Int. 6(6), pp. 721-741, 1984.

[27] H. Szu, and R. Hartley, "Fast simulated annealing," Phys. Lett. A, Vol.122, pp. 157-162, 1987.

[28] L. Ingber, "Very fast simulated re-annealing," Mathl. Comput. Modelling, Vol.12, pp. 967-973, 1989.

[29] A. Basu, and N. Frazer, "Rapid determination of the critical temperature in simulated annealing inversion," Science, Vol.249, pp. 1409-1412, 1990.



Name:
Shen Furao

Affiliation:
Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

Address:
R2-52 Hasegawa Research Group, Imaging Science and Engineering Lab., 4259 Nagatsuta, Midori-ku, Yokohama 226-8503, Japan

Brief Biographical History:

1998- Lecturer at Nanjing University, China

2000- Senior researcher in Foursis Inc., Japan

Main Works:

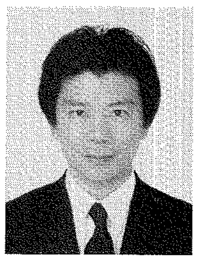
• S. Furao, and O. Hasegawa, "A fast no search fractal image coding method," Signal Processing: Image Communication, Vol.19, pp. 393-404, 2004.

• S. Furao, and O. Hasegawa, "A Self-organized Growing Network for On-line Unsupervised Learning," IEEE International Joint Conference on Neural Networks (IJCNN2004), CD-ROM ISBN:0-7803-8360-5, Vol.1, pp. 11-16, 2004.

Membership in Learned Societies:

• The Institute of Electrical and Electronics Engineers (IEEE)

• The Institute of Electronics, Information, and Communication engineers (IEICE)



Name:
Osamu Hasegawa

Affiliation:
Associate Professor,Imaging Science and Engineering Lab., Tokyo Institute of Technology

Address:
4259 Nagatsuta, Midori-ku, Yokohama, Kanagawa 226-8503, Japan

Brief Biographical History:

1993- Joined Elecytotechnical Laboratory (ETL)

1999- Visiting Scientist, Carnegie Mellon University (CMU)

2000- Advanced Industrial Science and Technology (AIST)

2002- Joined Tokyo Institute of Technology (TITech)

2002- PRESTO, Japan Science and Technology Agency (JST)

Main Works:

• N. Enomoto, T. Kanade, H. Fujiyoshi, and O. Hasegawa, "A method for monitoring activities of multiple objects by using stochastic model," IEICE Trans. on Information and Systems, Vol.84-D, No.12, pp. 1705-1712, 2001.

• B. Raytchev, O. Hasegawa, and N. Otsu, "User-independent online gesture recognition by relative motion extraction," Pattern Recognition Letters, 21 (1), pp. 69-82, 2000.

• S. Nobe, S. Hayamizu, O. Hasegawa, and H. Takahashi, "On lisetener's Fixation and Comprehension of Hand Gestures," Gestures. – Meaning and Use, Fernando Pessõa University Press, pp. 359-363, 2003.

Membership in Learned Societies:

• The Institute of Electrical and Electronics Engineers (IEEE)

• The Institute of Electronics, Information, and Communication engineers (IEICE)

• The Information Processing Society of Japan (IPSJ)

• The Japan Society of Artificial Intelligence (JSAI)

• The Japanese Cognitive Science Society (JCSS)