

Short Paper

Adaptive Schemes for Improving Fractal Block Coding of Images

HSUAN T. CHANG AND CHUNG J. KUO*

*Department of Electronic Engineering
Chien Kuo College of Technology and Commerce
Changhua City, Taiwan 500, R.O.C.*

**Signal and Media (SAM) Laboratory
Department of Electrical Engineering
National Chung Cheng University
Chiayi, Taiwan 621, R.O.C.*

In this paper, we propose several adaptive methods to improve the fractal block coding (FBC) of still images. First of all, the range of the contrast scaling is adaptively selected according to its statistical property; then, we search for the best quantized value in the selected range. Then, the domain pool of the current range block is predicted using a finite-state method, which uses the information of previous best-match domain blocks and domain pools. The size of the domain pool is automatically adjusted; however, conventional studies can only determine a fixed-size domain pool. Therefore, the search process for finding the best-match domain block in our approach is more flexible. In addition, we propose a finite-state method to estimate the mean (average pixel value) of the current range block by using the means of the previous blocks. From the simulation, we achieve 0.35 dB improvement in peak signal-to-noise ratio and 10.5% reduction in bit rate on average compared with the basic FBC scheme. Finally, the encoding and decoding time of different FBC schemes is also evaluated for completeness.

Keywords: fractal block coding, range block, domain block, domain pool, finite-state method

1. INTRODUCTION

Fractal image coding is a new technique for image compression that has been successfully developed to automatically convert a gray-scale image into a set of contractive affine transformations (CATs) [1,2]. In the original two-level (*parent* and *child*) fractal block coding (FBC) scheme proposed by Jacquin, an image is automatically encoded by mapping the contractive-affine-transformed domain block D (with size $B \times B$) to the range block R (with size $B/2 \times B/2$). Then, the parameters (called the *fractal code*) describing the CAT which has the minimum matching error are transmitted or stored. The fractal code includes the contrast scaling α , luminance shift Δg (or the block mean μ_R [3]), isometry I , and the position of the best-match domain block, P_D , in the domain pool D .

Received May 28, 1997; accepted May 12, 1998.
Communicated by Soo-Chang Pei.

We propose the adaptive schemes based on a modified FBC scheme, which is called the basic method here. In this basic method, the luminance shift is replaced by the block mean (average pixel value), and the domain pool is fixed and overlays the range block. On the other hand, the determination of contrast scaling and the isometry operation are the same as those in the original FBC scheme. The proposed adaptive methods offer further improvements based on this basic method.

First of all, we adaptively determine the range of contrast scaling and then search for the best quantized value in the selected range. A finite-state method is proposed to select the domain pool for the child range block. There are three domain-pool sizes in our design. For each child range block, the corresponding domain pool is selected using the next-state function by means of the positions of two previous best-match domain blocks and the sizes of corresponding domain pools. Note that in this design, a variable-size domain pool is proposed for the first time for the FBC scheme since the design of the conventional domain pool only allows a fixed size to be chosen. On the other hand, the block mean is also determined using a next-state function. We measure the difference between the maximum and minimum means of the blocks near the current range block. According to this difference, the range and number of the bits required to denote the mean are adaptively determined. Simulation results show that based on the proposed methods, we obtain better image quality and a higher compression ratio compared to the basic FBC scheme. Compared with other fractal schemes, the proposed scheme also has superior performance. Finally, the encoding and decoding time of the basic and adaptive schemes will also be evaluated for completeness.

The organization of this paper is as follows: The modified FBC scheme that serves as the basic method is described in Section 2. In Section 3, an adaptive method to select the range of contrast scaling and its step size of quantization is presented at first. Then two finite-state schemes to determine the child domain pool and the mean of the range block are described. The simulation results are shown in Section 4 and finally we give a conclusion in Section 5.

2. BASIC FBC SCHEME

We propose the adaptive schemes based on a modified FBC scheme, which is called the basic method here. In this basic method, the sizes of the range and domain blocks are the same as those in the original FBC scheme. However, we discard the block classification, and the domain pool consists of the 64 domain blocks nearest to the range block; thus, the position of the domain block is coded using six bits. The luminance shift is replaced by the mean [3], which is coded by six bits here. The contrast scaling is quantized to the nearest value in the set $\{n/8, n = 1, 2, 3, \dots, 8\}$ and coded by three bits. On the other hand, the eight isometries are the same as those in [2] and coded by three bits.

The coding configuration of the parent range block in this basic scheme is different from that of the original FBC scheme. A variable-length header is attached to the fractal code to identify the partition of the range block. Table 1 shows the header and the bit allocation for the different partitions of the range block. We assign '0' as the header of the mean-coded parent range block. For the parent range block coded by CAT, '10' is the header. The header '11' means that a parent range block is split into four child blocks. Then, '0' and '1' represent the child range block coded by the mean and CAT, respectively. Therefore, the header has various lengths (one, two, and six bits) for different parent range blocks. Based on this basic FBC scheme, we propose several adaptive schemes, which are shown in the following section.

Table 1. Header and bit allocation for both parent and child range blocks.

| Block type | Header | Bit allocation | | | |
|-----------------------------|--------|----------------|---------|----------|-------|
| | | I | μ_R | α | P_D |
| R_p coded by μ_R | '0' | | 6 | | |
| R_p coded by CAT | '10' | 3 | 6 | 3 | 6 |
| R_p split into four R_c | '11' | | | | |
| R_c coded by μ_R | '0' | | 6 | | |
| R_c coded by CAT | '1' | 3 | 6 | 3 | 6 |

3. ADAPTIVE SCHEMES FOR FBC OF IMAGES

3.1 Determination of Contrast Scaling

Here, we propose two methods to improve the determination of the contrast scaling. We first find that the pixel values in image blocks are seldom uniformly distributed within the dynamic range. Hence, the method of determining the contrast scaling in [2] may not be optimal because the matching error corresponding to the determined contrast scaling is not minimized. Therefore, in order to find the best contrast scaling that minimizes the matching error, we can determine the contrast scaling by searching all the possible quantized values in the defined range. With this searching process, we can ensure that a minimum matching error or a minimum quantization error [15] is achieved.

As mentioned in [16], under some conditions, the convergence property can be maintained when the contrast scaling is greater than one. Also in [17], for some range blocks, contrast scaling greater than one can achieve minimum distortion between the range block and the transformed domain block. Our experimental data shows that some of the contrast scalings greater than one lead to smaller matching distortions. Therefore, we next adaptively adjust the range of contrast scaling such that its maximum value can be greater than one to reduce the matching error. The contrast scaling has to be quantized into one of the eight values in a selected range. In order to decide on the best range and step size for the quantization, we use the optimal uniform quantization for nonuniform distribution [4]. A probability density function (pdf) of the contrast scaling at the parent level is measured at first. This pdf is approximated by a nonuniformly distributed function with mean μ_α and standard deviation σ_α . Then, the range and step size for quantizing the contrast scaling are determined according to the pdf-optimized method given in [4]. The optimal step size is given by

$$\Delta\alpha_{opt} = \lambda \cdot \sigma_\alpha, \quad (1)$$

where λ is a constant and related to which nonuniform distribution is chosen and how many quantization levels are used. The selected range $\alpha_{min} \sim \alpha_{max}$ is given by

$$\begin{cases} \alpha_{min} = \mu_\alpha - \frac{2^v - 1}{2} \cdot \Delta\alpha_{opt} \\ \alpha_{max} = \mu_\alpha + \frac{2^v - 1}{2} \cdot \Delta\alpha_{opt}, \end{cases} \quad (2)$$

where v is the number of bits for denoting the contrast scaling.

After the range and step size of quantization at the parent level are decided, we repeat the same procedures to obtain the range and step size of quantization at the child level. In this method, the range of contrast scaling should be transmitted to the decoder. However, only a few extra bits are required to denote this information, hence, the increase in the bit rate is negligible. Simulation results show that the performance is improved due to these modifications with respect to contrast scaling.

3.2 Finite-State Domain Pool Design

Let the top-left corner of the domain block denote its position and the central domain block of the domain pool be the one at whose center the range block is located. For a range block, the position of its best-match domain block can be represented by the displacements (m_x, m_y) in the x and y directions from the center position of the domain pool. After investigation of the position distribution of the best-match domain block, the center position shows a slightly higher probability among all the positions, and others have equally like probabilities [8-11]. Some methods [5-10] have been proposed to design the fixed-size domain pool that covers the current range block. However, a large domain pool is inefficient when the position of the best-match domain block is located near the center position. More computations will be wasted in finding the best-match domain block, and more bits will be needed to denote the position while comparing with a smaller domain pool. Therefore, in order to efficiently search for the best-match domain block, it is desirable to have a domain pool whose size can be adaptively adjusted according to the information given by previously coded blocks.

In the two-level FBC scheme, the variance of a parent range block R_p with the size $B \times B$,

$$\text{Var}\{R_p\} = \frac{1}{B^2} \sum_{0 \leq a, b < B} (r_{a,b} - \mu_{R_p})^2, \quad (3)$$

is measured (where $r_{a,b}$ denotes the (a, b) th pixel in the parent range block and μ_{R_p} denotes the block mean of a parent range block) and compared with a threshold value E_{th} . If $\text{Var}\{R_p\} \leq E_{th}$, the parent range block is coded by mean. Otherwise, the parent range block is coded by CAT. The same threshold value is applied to determine whether or not a parent range block is split into four child range blocks. If the mean-squared-error (MSE) measurement of the parent range block and coded parent range block \hat{R}_p ,

$$\text{MSE}(R_p, \hat{R}_p) = \frac{1}{B^2} \sum_{0 \leq a, b < B} (r_{a,b} - \hat{r}_{a,b})^2, \quad (4)$$

is smaller than or equal to the threshold value (*i.e.*, $\text{MSE}(R_p, \hat{R}_p) \leq E_m$), then the parent range block is coded by CAT. Otherwise, the parent range block is split into four child range blocks for further processing.

Because the child range blocks have stronger correlations than the parent range blocks, the proposed finite-state method is only used to determine the child domain pool. For a child range block whose variance is greater than a predefined threshold value, we proceed with CAT to find its fractal code. Before starting the matching process, we have to determine the corresponding domain pool. The block diagrams of the proposed adaptive scheme for

determining the child domain pool in the encoder and decoder are shown in Figs. 1(a) and 1 (b), respectively. In this method, the domain pool is adaptively determined by the proposed next-state function, whose arguments are the positions of two previous best-match domain blocks, (P_{D_i}, P_{D_j}) and corresponding domain pools (D_i, D_j) . In the encoder, the best-match domain block is then obtained by searching all the domain blocks in the determined domain pool. On the other hand, we find the best-match domain block from the fractal code and reconstruct the coded range block \hat{R}_c in the decoder.

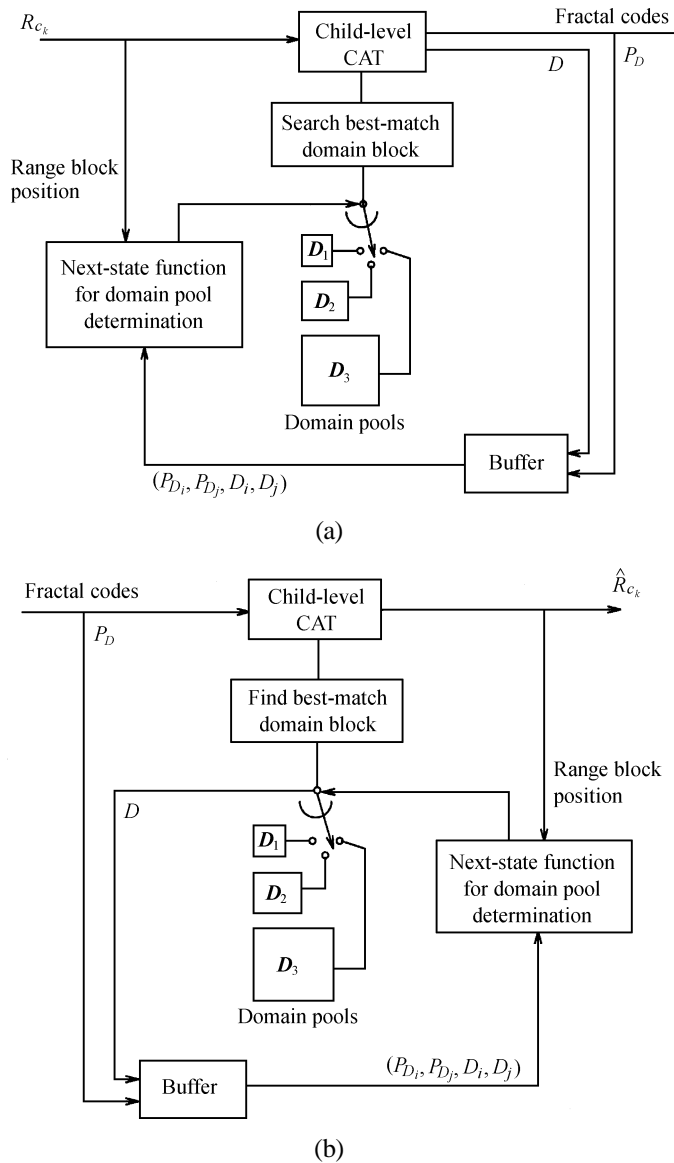


Fig. 1. The block diagram of the proposed finite-state method for domain pool determination in: (a) encoder, (b) decoder.

From the statistical properties shown in [10,11], we know that the domain blocks near the range block have a slightly higher probability of obtaining lower MSEs. On the other hand, other domain blocks have equally like probabilities. Therefore, we let the position of the domain block be nonuniformly distributed in all the positions of the domain pool. There are three sizes in our domain pool design: D_1 , D_2 , and D_3 . Fig. 2(a)-(c) show the three domain pools $D_1 \sim D_3$, which consist of 16, 32, and 64 domain blocks, respectively. Each domain block is represented by a dot which denotes the top-left corner of the domain block. In D_1 and D_2 , the 13 domain blocks nearest to the center position are separated by one pixel either horizontally or vertically, and the others have two-pixel separation. On the other hand, the outer domain blocks (except the domain blocks in D_2) in D_3 have three-pixel separation. Based on this design, our next-state function for determining the child domain pool is proposed and described below.

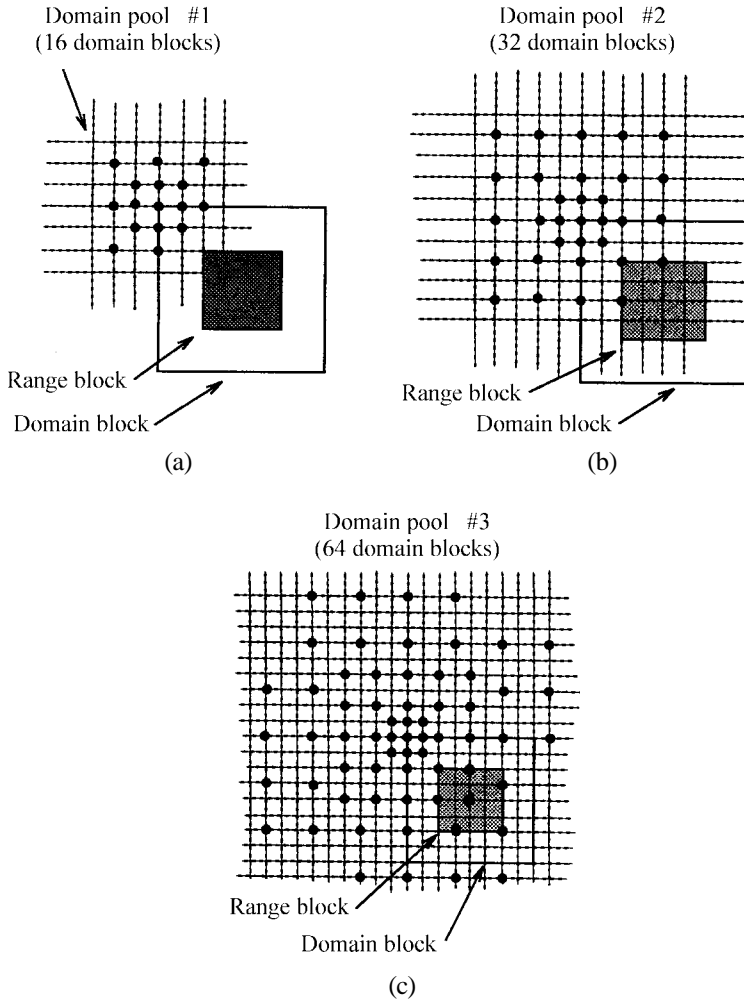


Fig. 2. The proposed new domain pool design in our adaptive FBC scheme: (a) D_1 : 16, (b) D_2 : 32, and (c) D_3 : 64 domain blocks.

For a child range block R_{c_k} , let D_i and D_j denote the domain pools corresponding to the upper and left range blocks R_{c_i} and R_{c_j} , respectively. The idea of the next-state function design is based on that if the best-match domain block of the range block R_{c_i} or R_{c_j} is far from the center of the domain pool, then the current domain pool D_k should be extended. Otherwise, the current domain pool should be contracted or, at least, kept at the same size. Now, the domain pool is not fixed, and it is adaptively changed according to the information of the previous best-match domain blocks and domain pools.

The proposed next-state function $\eta(P_{D_i}, P_{D_j}, D_i, D_j)$ operates as follows: First we calculate the distances d_i and d_j from $P_D(m_{x_i}, m_{y_i})$ and $P_D(m_{x_j}, m_{y_j})$ to the center positions of the domain pools D_i , and D_j , respectively:

$$\begin{cases} d_i = \sqrt{m_{x_i}^2 + m_{y_i}^2} \\ d_j = \sqrt{m_{x_j}^2 + m_{y_j}^2} \end{cases} \quad (5)$$

The child domain pool D_k for the range block R_{c_k} is determined using the procedures described in the following pseudo codes:

{The Pseudo Codes for Domain Pool Determination}

Let $D_m = \text{maximum}(D_i, D_j)$, $d_m = \text{maximum}(d_i, d_j)$

if (R_{c_k} is at the boundary of the image) $D_k = D_0$;

else

if ($D_m = D_1$)

if ($d_m > t_1$) $D_k = D_2$; else $D_k = D_1$;

if ($D_m = D_2$)

if ($d_m < t_2$) $D_k = D_1$; else if ($d_m > t_3$) $D_k = D_3$; else $D_k = D_2$;

if ($D_m = D_3$)

if ($d_m < t_4$) $D_k = D_2$; else $D_k = D_3$.

Here, D_0 denotes the domain pool used in the basic FBC scheme. In the above codes, four decision factors $t_1 \sim t_4$ are used to control the numbers of domain pools with different sizes. This will affect the fidelity and the bit rate of the decoded image. The choice of decision factors is related to the maximum distance d_{max} from the farthest position of the domain block to the center of the domain pool. For example, if both t_1 and t_2 are close to d_{max} , then the number of the domain pool D_1 will increase and the bit rate decrease. On the other hand, if both t_3 and t_4 are much less than d_{max} , then the number of the domain pool D_3 will increase and the bit rate increase, also. Note that the value of t_3 can not be less than t_2 in our algorithm.

If the neighboring blocks of the range block R_{c_k} are not coded by CAT (*i.e.*, the parent range is block coded by mean or CAT, or child range block coded by mean), we use following method: For a parent range block coded by CAT, we directly apply the parent-level parameters $\{P_{D_i}, P_{D_j}, D_i, D_j\}$ to the next-state function. Otherwise, these parameters are reset to the default values, $D_i = D_j = D_3$ and $P_{D_i} = P_{D_j} = (4, 4)$, for the mean-coded parent and child range blocks.

3.3 Mean Estimation for Range Blocks

Usually the means of the neighboring blocks are similar. We can thus predict an adequate range for the block mean by using the means of previous range blocks and then code this block mean using a finite-state method. Since the predicted range is selected by one of the defined ranges, the number of bits required to represent the block mean in the various range is now changeable.

Two similar next-state functions for parent and child range blocks are designed to estimate the range of the block mean. Considering a parent range block, we can find six child range blocks $R_{c_A} \sim R_{c_F}$ on its west, north, north-west, and north-east sides (see Fig. 3(a)). Their means $\mu_{c_A} \sim \mu_{c_F}$ are used in our next-state function design. We first calculate the difference δ_p between the maximum and minimum values of these means:

$$\delta_p = \max(\mu_{R_{c_A}}, \dots, \mu_{R_{c_F}}) - \min(\mu_{R_{c_A}}, \dots, \mu_{R_{c_F}}). \quad (6)$$

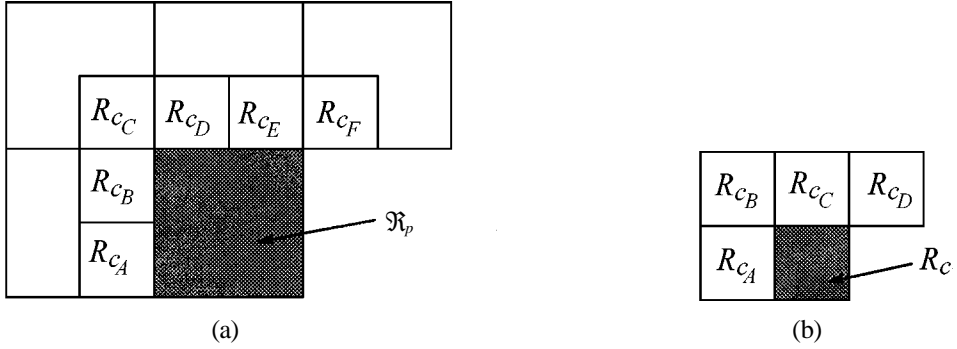


Fig. 3. Blocks for estimating the range of the mean of the current (a) parent and (b) child range blocks.

The relationship between the difference δ_p and the estimated range \mathfrak{R}_p is as follows: For the parent range block coded by mean,

$$\mathfrak{R}_p = \begin{cases} 0 \sim 252 \text{ (6 bits)}, & \text{if } \delta_p \geq 64 \\ \mu_{ct} - 62 \sim \mu_{ct} + 62 \text{ (5 bits)}, & \text{if } 64 > \delta_p \geq 32 \\ \mu_{ct} - 30 \sim \mu_{ct} + 30 \text{ (4 bits)}, & \text{if } 32 > \delta_p \geq 16 \\ \mu_{ct} - 14 \sim \mu_{ct} + 14 \text{ (3 bits)}, & \text{elsewhere,} \end{cases} \quad (7)$$

where the step size of quantization for the block mean is four; μ_{ct} is calculated by averaging the six mean values and serves as the central value of the determined range. For the parent range block coded by CAT,

$$\mathfrak{R}_p = \begin{cases} 0 \sim 252 \text{ (6 bits)}, & \text{if } \delta_p \geq 32 \\ \mu_{ct} - 62 \sim \mu_{ct} + 62 \text{ (5 bits)}, & \text{if } 64 > \delta_p \geq 16 \\ \mu_{ct} - 30 \sim \mu_{ct} + 30 \text{ (4 bits)}, & \text{if } 32 > \delta_p \geq 8 \\ \mu_{ct} - 14 \sim \mu_{ct} + 14 \text{ (3 bits)}, & \text{elsewhere,} \end{cases} \quad (8)$$

Note that in Equations (9) and (10), if the upper bound (lower bound) of the determined range is greater than 252 (lower than 0), then it is shifted to 252 (0). Basically, a larger difference leads to a larger range and more representative bits (and vice versa). Therefore, the block mean is coded within this predicted range, and the required number of bits is determined at the same time.

For a child range block, we can find four child range blocks on its west, north, north-west, and north-east sides (see Fig. 3 (b)). Therefore,

$$\delta_c = \max(\mu_{R_{c_A}}, \mu_{R_{c_B}}, \mu_{R_{c_C}}, \mu_{R_{c_D}}) - \min(\mu_{R_{c_A}}, \mu_{R_{c_B}}, \mu_{R_{c_C}}, \mu_{R_{c_D}}). \quad (9)$$

The mapping from the difference δ_c to the range \mathfrak{R}_c for the current child range block is similar to Equation (7) except that δ_p is replaced by δ_c . Finally, the mean is determined and restricted within the estimated range. The number of representative bits is now adaptively changed according to the next-state function. This is more efficient because we exploit the spatial continuity of the block mean. Therefore, more savings in the bit rate based on this design is expected.

4. COMPUTER SIMULATION AND DISCUSSION

In our computer simulation, four 512×512 images (Lena, Building, Harbour, and Jetplane) with eight-bit gray-scale resolution were used to test the proposed adaptive FBC schemes. Fig. 4(a)-(d) show these four original images. The performance of the decoded image quality is represented by the peak signal-to-noise-ratio (PSNR), which is defined as

$$\text{PSNR} = 10 \log_{10} \frac{512^2 \cdot 255^2}{\text{B}^2 \sum_{i=1}^{N_p} \text{MSE}(R_{p_i}, \hat{R}_{p_i}) + \frac{\text{B}^2}{4} \sum_{i=1}^{N_c} \text{MSE}(R_{c_i}, \hat{R}_{c_i})} \quad \text{dB}, \quad (10)$$

where N_p and N_c are the total number of parent (8×8) and child (4×4) range blocks, respectively. In the simulation, we let the threshold values E_{th} at the parent and child levels be 25.

The criterion for the decoded image to achieve convergence is determined as follows: Let the n th iterated image be denoted by $f^{(n)}$. The average error $e(n)$ between the n th and $(n-1)$ th decoded images is calculated by

$$e(n) = \frac{1}{512^2} \sum_{i=1}^{512} \sum_{j=1}^{512} (f_{i,j}^{(n)} - f_{i,j}^{(n-1)})^2, \quad (11)$$

where $f_{i,j}^{(n)}$ denotes the (i,j) th pixel in n th decoded image. If the ratio

$$\gamma = \frac{|e(n) - e(n-1)|}{e(n-1)} \quad (12)$$

is smaller than a threshold value γ_{th} , then the decoded image converges and the iteration

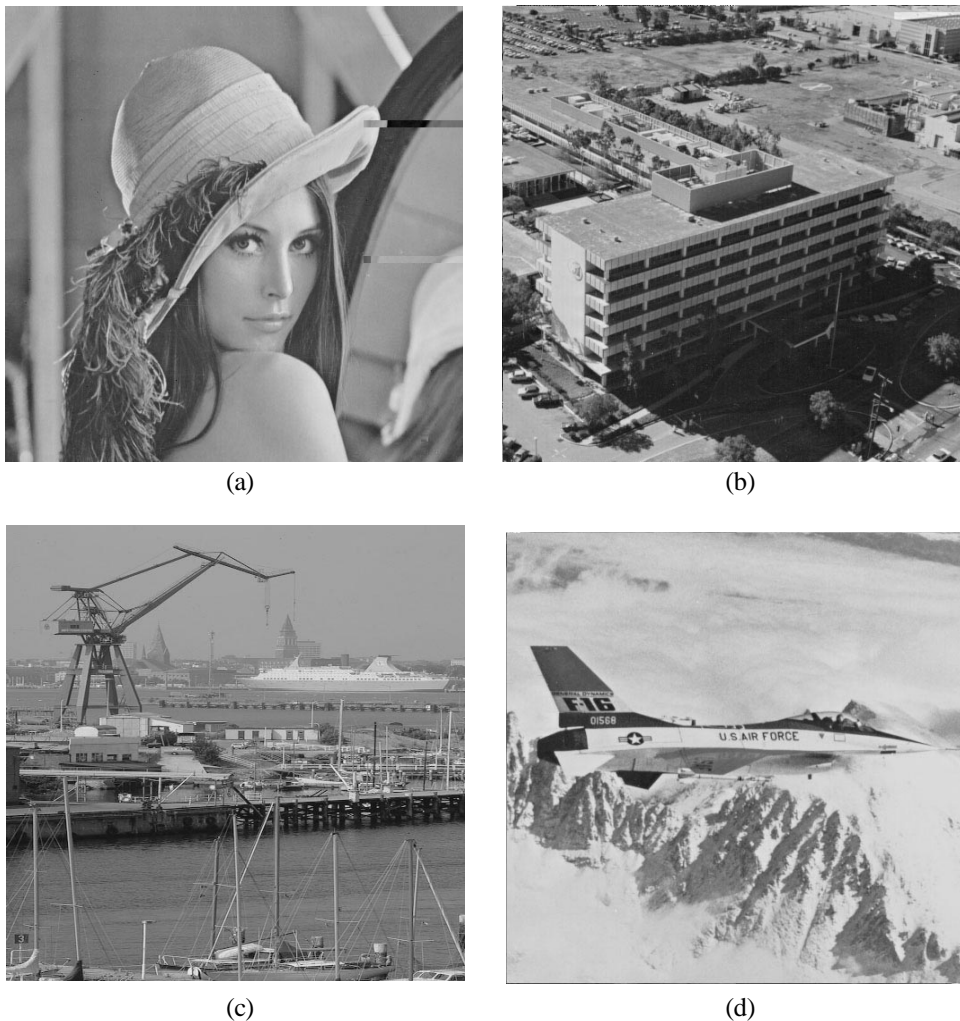


Fig. 4. Four test images used in the proposed adaptive FBC scheme: (a) Lena, (b) Building, (c) Harbour, (d) Jetplane.

process terminates. Otherwise, the iteration process will not stop until the criterion $\gamma \leq \gamma_{th}$ is satisfied. Here, the convergent criterion γ is set by 0.003. The performance of the basic FBC scheme is shown in Table 3. We will compare the proposed adaptive methods with the basic FBC scheme and other existing fractal coding schemes.

To determine the range of contrast scaling, we first investigated the distribution of the experimental data (0.1~2, step by 0.1) at the parent level. Considering the Lena image, its distribution of experimental data is shown in Fig. 5(a). Some contrast scalings greater than one give the minimum MSE; thus, the range of contrast scaling should not be fixed to 0~1. We then fit the curve shown in Fig. 5(a) to a normal distribution with mean 0.57 and variance 0.038. Using the uniform quantizer for nonuniform distribution shown in [6], the optimal step

size and the range were calculated using Equations (3) and (4), respectively. Here $\lambda = 0.586$ and $\nu = 3$; thus we could obtain step size of 0.11 and a range of 0.19~0.96. For the other three images, we chose their step sizes and ranges (shown in Table 2) in the similar way.

Once the range and step size for quantizing the contrast scaling at parent level were decided, we next investigated the distribution of the experimental data at the child level. This distribution for the Lena image is shown in Fig. 5(b). We could fit these data by a normal distribution with mean 0.70 and variance 0.075 to obtain an optimal step size of 0.16 and a range of 0.14~1.26. Using on the same method, we could choose the step sizes and ranges for the other three images. The step sizes and ranges of the contrast scaling at the child level for the four test images are summarized in Table 2. After the step size and range of the contrast scaling were decided, all eight possible values of contrast scaling were tested to find the best one to minimize the MSE. The simulation results are shown in Table 3. In this table, the PSNR and bit rate are both improved due to this adaptive selection for contrast scaling. We then performed the following steps based on this adaptive determination of the contrast scaling.

We used the proposed finite-state method to determine the domain pool for each range block. The maximum distance d_{max} for each domain pool was calculated first. As shown in Fig. 2, the maximum distances for the domain pools D_1 , D_2 , and D_3 were 2.83, 5.66,

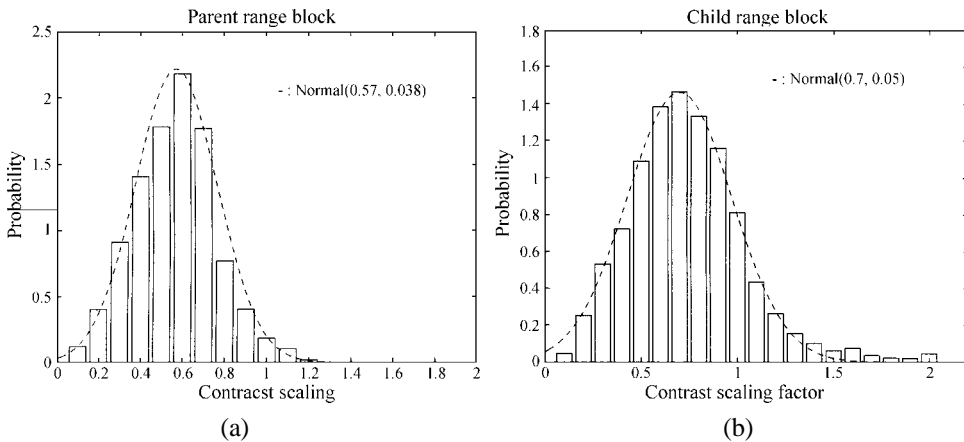


Fig. 5. The statistical properties of the experimental data of the contrast scaling for the Lena image: (a) Parent range block, (b) Child range block.

Table 2. The range and step size of quantization of the contrast scaling for each test image.

| | Image | Lena | Building | Harbour | Jetplane |
|--------------|----------------------|-----------|----------|-----------|-----------|
| 8×8 | Range of α | 0.19~0.96 | 0.1~1.15 | 0.15~1.27 | 0.15~1.27 |
| | $\Delta\alpha_{opt}$ | 0.11 | 0.15 | 0.16 | 0.16 |
| 4×4 | Range of α | 0.14~1.26 | 0.2~1.39 | 0.25~2.0 | 0.15~1.34 |
| | $\Delta\alpha_{opt}$ | 0.16 | 0.17 | 0.25 | 0.17 |

and 10.44, respectively. We thus chose the decision factors $t_1 = t_2 = 2$, $t_3 = 3$ and $t_4 = 5$ so that the bit rate would not significantly increase. With the simulation, we could obtain a decoded by 6 mi²TO andtion,uld not is-reducedby ect5%ton averTji. 10 32.418 .009 Tci²AND ablei5.0.013 Tci²-0.018 Twi²[(= 6 K)](Obvious. ,,our Tdap W) method,shows good,fidelity,oftion,i²T*1²- t

Table 4. Computation time (in seconds) for the basic and proposed adaptive FBC schemes.

| Image | | Basic FBC | α | $\alpha + D$ | $\alpha + \mu_R$ | Combined |
|----------|--------|-----------|----------|--------------|------------------|----------|
| Lena | Encode | 110 | 519 | 580 | 570 | 638 |
| | Decode | 19 | 5 | 5 | 5 | 6 |
| Building | Encode | 201 | 947 | 1050 | 963 | 1067 |
| | Decode | 18 | 8 | 9 | 9 | 9 |
| Harbour | Encode | 178 | 844 | 1007 | 857 | 1036 |
| | Decode | 22 | 8 | 9 | 9 | 9 |
| Jetplane | Encode | 115 | 534 | 587 | 546 | 596 |
| | Decode | 11 | 5 | 6 | 5 | 6 |

Table 5. Performance comparison of Lena image between the adaptive FBC scheme and other fractal coding schemes.

| Scheme | Saupe [14] | Fisher [11, Chap. 6] | Vines [11, Chap. 10] | Lu [12] | Bogdan [13] | Proposed |
|----------------------|------------|----------------------|----------------------|---------|-------------|----------|
| PSNR (dB) | 33.0 | 34.62 | 30.5 | 32.17 | 33.08 | 34.06 |
| Bit rate (bit/pixel) | 0.42 | 0.54 | 0.44 | 0.62 | 0.81 | 0.43 |

5. CONCLUSIONS

In this paper, we have modified the original FBC scheme; this modified scheme is called the basic method. Based on this basic method, we have proposed several adaptive methods to improve the FBC scheme. First of all, we adaptively select the range of the contrast scaling and search for the best quantized value in the selected range. The domain pool is determined using the next-state function, which employs information about the previous best-match domain blocks and domain pools. Conventional studies can determine the domain pool with a fixed size. The size of the domain pool is automatically adjusted; hence, this design is more efficient. In addition, we exploit spatial redundancy in the neighboring range blocks; hence, we can reduce the bit rate. Using the computer simulation, the proposed adaptive FBC scheme produces decoded images whose PSNR is improved by 0.35 dB and whose bit rate is reduced by 10.5% on average. The encoding and decoding time for the proposed adaptive schemes is also evaluated. From the simulation results, it has been found that our adaptive FBC scheme is attractive and superior compared with the existing fractal schemes.

Recently, the FBC scheme has been shown to have deep connections with other block-based image coding techniques, such as transform coding, vector quantization, and wavelet coding. Our future work will focus on the extending the proposed adaptive schemes by employing above techniques to achieve further improvement.

ACKNOWLEDGMENT

The authors thank the reviewers for valuable comments and suggestions. This research was partially supported by the National Science Council, Taiwan, under contracts NSC 86-2221-E-194-009 and NSC 88-2612-E-270-001.

REFERENCES

1. A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, Vol. 1, No. 1, 1992, pp. 18-30.
2. A. E. Jacquin, "Fractal image coding: a review," *Proceedings of the IEEE*, Vol. 81, No. 10, 1993, pp. 1451-1465.
3. J. M. Beaumont, "Advances in block based fractal coding of still images," *IEE Colloquium on Application of Fractal Techniques in Image Processing*, 1990, pp. 3/1-3/5.
4. N. S. Jayant and P. Noll, *Digital Coding of Waveforms - Principles and Applications to Speech and Video*, Chapter 4, Prentice-Hall International Inc., Taiwan, 1984.
5. H. T. Chang and C. J. Kuo, "An improved scheme for fractal image coding," *IEEE 1995 International Symposium on Circuits and Systems*, Vol. 3, 1995, pp. 1624-1627.
6. H. T. Chang and C. J. Kuo, "Fractal block coding using simplified finite-state algorithm," in *Proceedings of SPIE's Symposium on Visual Communication and Image Processing*, Vol. 1, No. 3, 1995, pp. 536-544.
7. G. Vines and M. H. Hayes III, "Adaptive IFS image coding with proximity maps," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, 1993, pp. 349-352.
8. D. Monro, "Generalized fractal transform: Complexity issues," in *Proceedings of IEEE Data Compression Conference*, 1993, pp. 254-261.
9. K. U. Barthel, J. Schuttmeier, T. Voyer and P. Noll, "A new image coding technique unifying fractal and transform coding," in *Proceedings of International Conference on Image Processing*, Vol. 3, 1994, pp. 112-116.
10. S. J. Woolley and D. M. Monro, "Optimum parameters for hybrid fractal image coding," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, 1995, pp. 2571-2574.
11. Y. Fisher, *Fractal Image Compression: Theory and Applications*, Chapters 6 and 10, Springer-Verlag, 1995.
12. G. Lu and T. Yew, "Image compression using partitioned iterated function system," *Proceedings of SPIE*, Vol. 2186, 1994, pp. 122-132.
13. A. Bogdan, "The fractal pyramid with applications to image coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, 1995, pp. 2595-2598.
14. D. Saupe, "The futility of square isometries in fractal image compression," in *Proceedings of IEEE International Conference on Image Processing*, Vol. 1, 1996, pp. 161-164.
15. G. Oien, "Parameter quantization in fractal image coding," in *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, 1994, pp. 142-146.
16. B. Hurtgen and S. Simon, "On the problem of convergence in fractal coding schemes," in *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, 1994, pp. 103-106.

