Entropy Improvement for Fractal Image Coder

Jyh-Horng Jeng, Shuo-Li Hsu, and Yukon Chang
Department of Information Engineering, I-Shou University, Taiwan

Abstract: *Fractal Image Coder (FIC) makes use of the self-similarity inside a natural image to achieve high compression ratio and maintain good image quality. In FIC, the most important factor affecting the compression ratio and the image quality is the quantization of the contrast scaling and brightness offset coefficients. Most quantization methods treat the two coefficients independently and quantize them separately. However, the two coefficients are highly correlated and scatter around a line. In this paper, a joint coefficient quantization method is proposed that considers the two coefficients together and thereby achieves better compression ratio and image quality. The proposed method is especially effective under parsimonious conditions. For example, using only 3bits each to represent the contrast and brightness coefficients of Lena, the proposed method yields quality of 27.04dB, which is significantly better than 22.87dB obtained from the traditional linear quantization method.*

Keywords: *FIC, dihedral transformation, entropy, quantization, contrast adjustment, brightness offset.*

Received March 9, 2010; accepted October 24, 2010

1. Introduction

A fractal is a geometric shape that can be divided into parts each of which is a size-reduced copy of the origin subject to possible rotation and translation [11], a property called self-similarity. Fractal Image Coder (FIC) [4], a technique introduced by Barnsley and Demko [1], is based on the idea that image redundancies can be removed by means of exploiting local self-similarities in the image and representing them through affine transformations [14]. Due to the merit in mathematics and interesting phenomenon, fractal-based image processing has received a great deal of attentions including robust image compression [9], image retrieval [20], image recognition [10], and video compression [15]. During the encoding process, an image is partitioned into non-overlapping range blocks and a search for the best match in a domain pool is carried out for each of these range blocks. Due to the large size of the domain pool as well as the introduction of dihedral transformation, the encoding process is very time consuming. Consequently, a large portion of FIC-related researches are concerned with speeding up the encoder. Popular methods include genetic algorithms [20], one-norm of normalized block [2], classification method [11], prediction algorithm [12], and quadtree method [19]. Parallel architectures are also adopted to speedup the encoder [8].

The improvement of the encoding time is interesting since a great variety of methods can be applied. As a consequence, many researchers focus on how to speedup the encoder. Only few articles address the problem of compression ratio in which the idea of entropy is involved. In order to obtain a higher compression ratio, fractal code requires that the coefficients representing the contrast adjustment and brightness offset be represented using fewer bits. Quantization of these coefficients affects not only compression ratio but also the quality of the reconstructed image. Previously proposed coefficient quantization methods include the traditional linear coefficient quantization method, the Simulated Annealing (SA) optimization for near-optimal quantization [13], the Lloyd-Max quantizer [12], the LBG algorithm [5], and genetic algorithm [17]. Although contrast adjustment and brightness offset are to some extent linearly correlated, all except the LBG algorithm treat them as independent coefficients and quantize them separately. This results in worse image quality if the number of bits used is kept the same. On the other hand, the LBG algorithm, which uses vector quantization on the correlated coefficients to achieve better quality, suffers a lower compression ratio because it must store the coefficient index table for the given image. The term entropy, defined by Shannon [16], is a key related to data compression, which is commonly used to measure the compressibility of an image coder [7]. The entropy value of a data set stands for the theoretical minimal bits required to represent an indivisible unit, i.e., a symbol. On the other hand, as the mount of bits are pre-specified to represent a data set, we will prefer the entropy of this new data set be closing to that amount of bits so that we know we have already made full use of the bits. Analogous to image compression, the concept of entropy can also used to measure the capacity of text and image for information hiding systems [6].

In this paper, we propose a joint quantization method to represent the coefficients in FIC in order to make use of the pre-specified bits, in sense of entropy, as possible as we can. We first analyze the distribution pattern and the amount of occupied levels of the traditional linear quantization method. We then design a proper transformation so that we can fully make use of all the quantization points. We also calculate the amount of occupied levels and the entropy values in order to confirm the extent to which we have used the points. Experimental results show that the proposed method effectively reduces the number of bits consumed by the coefficients without degrading the image quality and outperforms the SA method [13] in efficiency.

2. Fractal Image Compression

The mathematical foundation of fractal image compression is the Iteration Function System (IFS) in which the governing theorems are the Contractive Mapping Fixed-Point Theorem and the Collage Theorem [4]. An ideal IFS hardly exists for a natural image because most of the sub-images are not directly similar to the whole image. To solve this problem, the idea of local self-similarity is adopted to form the Partitioned Iterated Function System (PIFS) [4].

Let f be a gray level image of size $N \times N$. Divide the image into a range pool of $(N/L)^2$ non-overlapping blocks of size $L \times L$. Let the contractility of the fractal coding be a fixed quantity of 2. The domain pool D is the set of all possible blocks of size $2L \times 2L$, which has $(N\text{-}16+1)^2$ elements. The amount of the domain blocks can be reduced by introducing the searching step size s, which is the distance in pixels between two adjacent domain blocks. Therefore, the reduced amount is $((N/s)\text{-}16+1)^2$. For each range block v, one searches in the domain pool to find the most similar domain block. To facilitate comparison, each domain block is first down-sampled so that it has the same size as the range block. For simplicity, let the terms domain block and domain pool refer to the down-sampled $L \times L$ blocks instead of the original $2L \times 2L$ blocks. The full search method transforms a domain block u using the eight transformations in the Dihedral group on the pixel positions to increase the size of the domain pool. If the origin of the coordinate of u is assumed to be located at the center of the block, the eight transformations, T_k, k=0, 1, ..., 7, are represented by the following matrices:

$$T_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ T_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},\ T_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},\ T_3 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$
$$T_4 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ T_5 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},\ T_6 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},\ T_7 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \quad (1)$$

The transformations T_1 and T_2 correspond to the flips of u along the vertical and horizontal lines respectively. T_3 is the flip along both the vertical and horizontal line. T_4, T_5, T_6 and T_7 are the transformations T_0, T_1, T_2 and T_3 followed by an additional flip along the main diagonal, respectively. Fractal coding also allows a contrast scaling, denoted by p, and a brightness offset, denoted by q, on the transformed domain blocks. Thus the fractal affine transformation Φ of $u(i, j)$ in D can be expressed as:

$$\Phi \begin{bmatrix} i \\ j \\ u_k(i,j) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & p_k \end{bmatrix} \begin{bmatrix} i \\ j \\ u_k(i,j) \end{bmatrix} + \begin{bmatrix} t_i \\ t_j \\ q_k \end{bmatrix} \quad (2)$$

where the 2×2 sub-matrix $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ is one of the dihedral transformations in equation 1 and (t_i, t_j) is the search entry in the image. Note that the domain blocks are regarded as being of size $2L \times 2L$.

Let u_k, k=0, 1, ...,7, denote the eight transformed blocks with u_0=u. At each search entry, eight separate MSE computations are required to find the index d such that:

$$d = \arg\min\{\mathrm{MSE}((p_k u_k + q_k), v) : k = 0,1,\cdots,7\} \quad (3)$$

where

$$MSE(u,v) = \frac{1}{L^2} \sum_{j=0}^{L-1} \sum_{i=0}^{L-1} (u(i,j) - v(i,j))^2 \quad (4)$$

The quantities p_k and q_k can be computed directly as:

$$p_k = \frac{L^2 < u_k, v > - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i,j) \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} v(i,j)}{L^2 < u_k, u_k > - (\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i,j))^2} \quad (5)$$

$$q_k = \frac{1}{L^2} (\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} v(i,j) - p_k \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i,j)) \quad (6)$$

where $<u_k, v>$ is the inner product. As u runs over all of the domain blocks in D, the one that matches v the best is found and the terms t_x and t_y in equation 2 are obtained. Together with d and the specific p and q corresponding to this d, the affine transformation equation 2 is constructed for the given range block v. A fractal code, from equation 2, includes five items: i, j, k, p, and q. The terms i and j represent the position of the best matched domain block, where i=t_x and j=t_y. Their bit representations are essentially decided by the sizes of the original image. For an image of size 256×256, 8bits are required to record each of i and j. There are eight isometric transformations as shown in equation 1. Therefore, 3bits are required to represent the coefficient k. The quantities p and q are the contrast adjustment and the brightness offset of a domain block, respectively, so the amount of levels quantizing them, i.e., bit representation, will affect the image quality directly. For an image of size 256×256 with 8×8 coding size, we typically allocate 5bits for p and 7 for q. If more bits are allocated, better image quality can be obtained but the compression ratio will increase as a result. The quantization of p and q is the main role controlling the rate-distortion tradeoff for fractal coding.

To retrieve an image from a fractal code, we first make up the $(N/L)^2$ affine transformations of the form equation 2 from the compression codes to constitute the PIFS. We choose an arbitrary image as the initial image and perform these affine transformations on it to obtain a new image. Then we perform the transformations on this image to obtain another new image. This process is repeated until the stopping condition is met. The Contractive Mapping Fixed-Point Theorem and the Collage Theorem guarantee that the sequence of images will converge. The stopping criterion is designed according to user's application and the final image is the retrieved image of the given fractal code.

3. Coefficient Quantization

As discussed in the previous section, the quantization of the coefficients p and q controls the rate-distortion tradeoff for fractal coding. We will now discuss the quantization in detail. A quantization function maps a continuous value c on the real axis R to a quantized level cj in $I=[L, U]$ where $j=1, ..., n$ and $n=2^m$ for some m. The quantization function Q is thus defined by:

$$Q : R \rightarrow \{c_1, c_2, \cdots, c_n\}$$

In practice, the function Q can be implemented through the definition of intervals partitioning $I=[L, U]$. We denote these intervals as $I_l=[L_l, U_l]$ and $I_j=[L_j, U_j]$ with $L_j=U_{j-1}$ for $j=2, 3, ..., n$ and calculate the quantized level c_j by:

$$c_j = \frac{U_j + L_j}{2}$$

If the real number c is out of the range of I, the corresponding minimum level c_1 or maximum level c_n will be used. The quantization function of the coefficients p and q can be defined accordingly, i.e.,

$$Q_p : R \rightarrow \{p_1, p_2, \cdots, p_n\} \text{ and } Q_q : R \rightarrow \{q_1, q_2, \cdots, q_n\} \quad (7)$$

The quantization may be defined by intervals of equal lengths. This is referred to as the linear quantization for which only the number of quantized levels need to be stored. Assume we adopt linear quantization for $I=[0, 1]$ and $n=4$. For such case, we do not need to record the levels c_j for $j=1, ..., 4$ and instead, only the number of levels, i.e., 4, is required to store since the quantized levels are equally spaced, which can be calculated by $c_j=j/4-1/8$. It may also be defined according to the clustering distributions, i.e., nonlinear quantization. In this case, the interval boundaries need to be saved together with the quantized levels, and sophisticated algorithm is required to find the proper boundaries. Among nonlinear quantization methods, Palazzari *et al*. [13] show that the SA optimization algorithm can achieve very good image quality. In their study, detailed discussion concerning nonlinear quantization outperforming linear quantization, including the underlying idea and derivations, is given. Essentially, the problem of quantization is viewed as an optimization problem, and from all of the possible quantization functions they pursue the optimal one.

In their terminology, let F denote the coding function representing the IFS code of the source image ζ, which is tied to the quantization functions $Q=[Q_p, Q_q]$ given in equation 7. Let Ω denote the reconstructed image in which the coefficients p and q are not quantized. For decoding from quantized p and q, we denote the retrieved image as $F(\zeta, Q)$ which is of course dependent on the quantization function Q. The relationship between the qualities affected by Q can be easily seen as:

$$d(\Omega, \zeta) < d(F(\zeta, Q), \zeta) = \Delta(Q)$$

where the function $d(., .)$ is a measure of distortion of two images. The search of quantization function Q is thus converted to the optimization problem:

$$\Delta(Q^*) = \min_{Q \in SQ} (\Delta(Q))$$

where SQ is the set of all possible quantization functions. The authors use SA to find a suboptimal solution that obtains better image quality in comparison to the linear quantization method. However, the computation is extremely extensive since thousands of fractal encodings are required to determine the quantization function. Their quantization method is so time consuming that they had to use supercomputers to speedup the overall encoding process.

4. The Proposed Coefficient Quantization Method

For fractal image coding, the contrast and brightness coefficients p and q are neither uniformly distributed nor uncorrelated on the p-q plane. Figure 1 plots all 1024 (p, q) points for the image Lena of size 256×256 with 8×8 range block. The original p and q are real numbers. The pairs (p, q) are plotted as small grey dots in Figure 1 as shown, the distribution of (p, q) points is highly non-uniform. Assuming p and q each uses 3bits, i.e., 8 levels for each and 64 quantization points in total, the quantization results are displayed in Figure 1 with triangular marks. As can be seen, only 23 (large dot) out of the 64 (triangle) quantization points are occupied. A more formal discussion of this phenomenon in terms of entropy is given below. The average amount of information of a set A with elements from the symbol set $\{s_0, s_1, ..., s_{m-1}\}$ defined by Shannon [16] is measured by the term entropy $H(A)$ defined as:

$$H(A) = \sum_{j=0}^{M-1} p_j \log_2(\frac{1}{p_j}) \quad (8)$$

where P_j is the probability of the symbol s_j appearing in the set A. It is well known that if each symbol has equal probability of appearance in the set A, the quantity $H(A)$ will attain the maximal value, i.e., 6, since 6bits are reserved to represent the data. In Figure 1, calculation according to equation 8 yields an entropy of 3.63, which is shown in Table 2 in the first row and the third column. This observation indicates that there is a rather large room for improvement if we can find a better way to quantize p and q so as to even out probabilities P_j over the entire set of A.

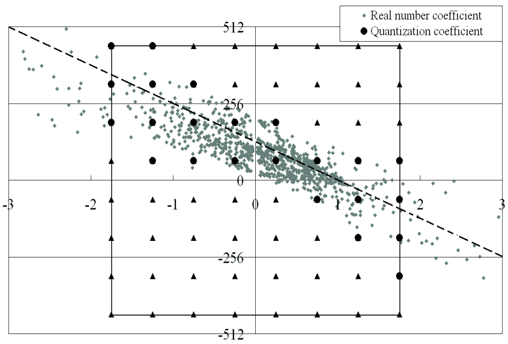


Figure 1. Distribution of p, q coefficients for Lena in the proposed linear quantization method with 3bits for each coefficient.

It is quite obvious from Figure 1 that the un-quantized (p, q) points are scattered around the line q=-128p+128 (dash line) on the p-q plane. In other words, p and q have the tendency of being negatively correlated more or less linearly. We therefore derive a transformation on the p-q plane followed by the traditional linear quantization according to the data distribution and the scattering tendency so as to increase the entropy. The transformation consists of a translation matrix followed by an axis rotation matrix defined as follows. For the sake of simplicity, we use the same symbols p and q for the axis notations during the intermediate steps and use p' and q' for the final step.

$$g_1(p,q) = \begin{bmatrix} 1 & 0 & \Delta p \\ 0 & 1 & \Delta q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \quad (9)$$

$$g_2(p,q) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \quad (10)$$

Specifically, the translation matrix is used first to shift the line q=-128p+128 to q=-128p with Δp=0 and Δp=-128. Next, the axis rotation matrix is applied to rotate the line q=-128p to q=0. The parameters $\sin\theta$ and $\cos\theta$ can be calculated from equations 11 and 12:

$$\cos\theta = \frac{ab}{\|a\|\|b\|} \quad (11)$$

$$\sin^2\theta + \cos^2\theta = 1\,v \quad (12)$$

We select two vectors a=(0, -1) and b=(1, -128)

$$\cos\theta = \frac{ab}{\|a\|\|b\|} = \frac{0\times(-1)+1\times-(128)}{\sqrt{0^2+(-1)^2}\sqrt{1^2+(-128)^2}} = \frac{-128}{\sqrt{16385}}$$

Since

$$\sin^2\theta + \cos^2\theta = 1$$

We have

$$\sin\theta = \sqrt{1-\cos^2\theta} = \sqrt{1-(\frac{-128}{\sqrt{16385}})^2} = \frac{1}{\sqrt{16385}}$$

The transformation therefore can be rewritten as follows:

$$g(p,q) = \begin{bmatrix} -\frac{128}{\sqrt{16385}} & \frac{1}{\sqrt{16385}} & 0 \\ -\frac{1}{\sqrt{16385}} & -\frac{128}{\sqrt{16385}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -128 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \quad (13)$$

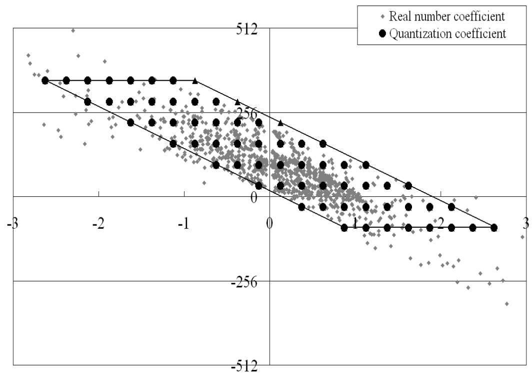


Figure 2. Distribution of p, q coefficients for Lena the proposed quantization method with 3bits for each coefficient.

In the p'- q' space, we restrict the values such that $p' \in [-1, 1]$and $q' \in [-256, 256]$. They are linearly quantized subsequently. The quantized values are then inverse-transformed back to the original p-q space to evaluate the similarity measure. The quantization results are shown in Figure 2, which is transformed back to the original space to facilitate the comparison with Figure 1. As shown in Figure 2, 61 (large dot) out of 64 quantized points (triangle) are occupied instead of 23 out of 64 in the traditional method shown in Figure 1. As illustrated, the traditional quantization is a rectangle while the proposed quantization exhibits a shape of parallelogram, which is designed according to the distribution of the original (p, q) pairs. The entropy calculated from our proposed method is 5.23, shown in the second row and third column of Table 2, which is quite close to the maximum value 6. It means that we have made efficient use of the 6bits available for p and q.

Table 2. Experimental results for Case 1 (PSNR, occupied levels, and entropy).

Bits for *p*, *q*		2, 2	2, 3	3, 3	3, 4	3, 7	4, 8	5, 7
Quantized Levels		16	32	64	256	1024	4096	4096
Lena	Linear	25.17	25.81	26.71	27.29	28.00	28.34	28.57
		5	11	23	37	200	451	447
		1.87	2.84	3.63	4.42	7.08	8.43	8.37
	Proposed	26.76	27.25	27.71	28.05	28.26	28.58	28.75
		16	28	61	104	438	731	726
		3.58	4.26	5.23	6.00	8.40	9.34	9.33
Baboon	Linear	18.93	18.47	19.42	19.66	19.85	19.95	20.03
		4	10	15	32	175	406	414
		1.92	2.51	3.26	4.01	6.79	8.25	8.31
	Proposed	19.45	19.63	19.80	19.89	19.93	19.99	20.08
		15	28	53	96	406	727	727
		3.11	3.97	4.86	5.79	8.26	9.34	9.32
F16	Linear	5.71	21.99	25.22	25.76	26.48	26.69	27.13
		4	12	154	221	128	303	293
		1.33	2.53	2.71	3.50	5.89	7.42	7.40
	Proposed	24.91	25.86	26.09	26.48	26.78	27.13	27.31
		13	17	35	56	253	487	474
		2.22	2.72	3.69	4.51	7.19	8.46	8.43
Pepper	Linear	26.18,	26.83	27.84	28.39	29.26	29.71	29.97,
		7	13	20	42	216	476	453
		2.09	2.92	3.78	4.67	7.16	8.55	8.48
	Proposed	27.86	28.47	28.92	29.32	29.56	30.00	30.22
		16	31	61	107	469	742	738
		3.65	4.40	5.29	6.11	8.53	9.38	9.35

5. Experimental Results

In this section, we show the experimental results of the proposed coefficient quantization method for FIC. We compare the qualities of reconstructed images between the SA method, the traditional linear coefficient method, and our method. The image quality is measured in terms of *PSNR*, given by:

$$PSNR = 10 \times \log_{10}(255^2 / \text{MSE}) \tag{14}$$

where *MSE* is given in equation 4. The experiment is implemented using Borland C++ Builder 6.0 running on a PC with Intel Core 2 Quad Q8400 2.66GHz CPU. In the encoding of Lena image of size 512×512 with coding size 8×8 and search step size 2, we use 3 bits to quantize each of p and q. Table 1 shows the results of linear, SA, and the proposed method. The linear method produces a PSNR of only 22.09dB image quality. Although the SA method successfully achieved a great improvement of 28.79dB, the proposed method obtains 29.60dB, which is even 0.8dB better than that of SA method. Moreover, the proposed method requires only 2.7 hours of running time on the PC stated above while the SA method needs 2.5 hours on an APE100/Quadrics supercomputer (peak computation power: 25.6 Gflops). Note that the executing times are almost the same for linear and proposed methods. Although the image qualities of the three methods are roughly the same as the number of bits used to quantize the coefficients increases, the proposed method still has the advantage of outperforming the other two methods for low bit rate cases.

Table 1. Image quality and time used using various methods for 512×512 Lena.

Bits for *p*, *q*	3, 3	3, 4	3, 7	4, 8	**Times**
Linear	22.09	24.84	29.69	30.22	2.7 h (PC)
SA	28.79	29.17	30.18	30.77	2.5 h (supercomputer)
Proposed	29.59	29.95	30.64	30.92	2.7 h (PC)

To further discuss the coding performance of the proposed method, we conduct the experiments on Lena, Baboon, F-16, and Pepper images, each of which is of size 256×256 with coding size 8×8 and search step size 1. We set up two quantization ranges:

- *Case 1:* $p \in [-2, 2]$, $q \in [-512, 512]$
  $p' \in [-1, 1]$, $q' \in [-256, 256]$
- *Case 2:* $p \in [-6, 6]$, $q \in [-1824, 1824]$
  $p' \in [-1, 1]$, $q' \in [-768, 768]$

The retrieved images from Case 1 are shown in Figure 3 in which there are 3bits for each of p and q. The first column consists of the original images. The second and the third columns are the retrieved images using linear method and the proposed method, respectively. The image qualities are almost the same in PSNR for both methods but the images of the proposed method in the third column are clearly visually superior to those of the linear method.

a1) Lena image. a2) Linear, PSNR= 26.71dB. a3) Proposed, PSNR =27.71dB.

b1) Baboon. b2) Linear, PSNR= 19.42dB. b3) Proposed, PSNR =19.80dB.

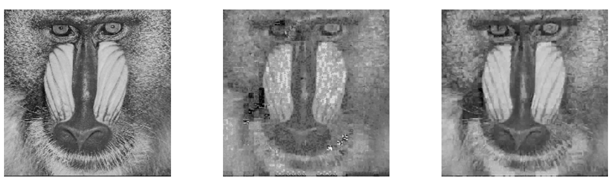
c1) F16. c2) Linear, PSNR= 25.22dB. c3) Proposed, PSNR =26.09dB.

d1) Pepper. d2) Linear, PSNR = 27.84 dB. d3) Proposed, PSNR =28.92dB.

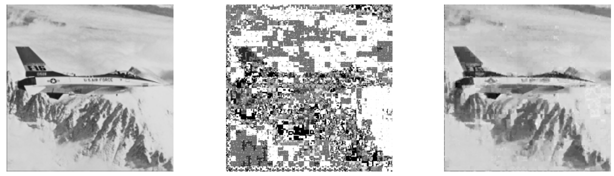
Figure 3. Coefficient quantization for Case 1.

In Case 2, the quantization ranges are larger. The coding results are shown in Figure 4 in which the linear method in the second column experiences a serious decay both in the PSNR measure and the visual effects. The PIFS even fails for F-16 as shown in Figure 4-c2. This problem is due to the heavily concentrated distribution of p and q in the $p-q$ plane, shown in Figure 5, so that only very few quantization points are visited. As can be seen from Figure 4, the proposed method in the third column still preserve good qualities as that in Case 1.

a1) Lena. a2) Linear, PSNR= 22.87dB. a3) Proposed, PSNR =27.04dB.



b1) Baboon. b2) Linear, PSNR= 17.92dB. b3) Proposed, PSNR =19.54dB.



c1) F16. c2) Linear, PSNR= 10.41dB. c3) Proposed, PSNR =25.71dB.

d1) Pepper. d2) Linear, PSNR= 19.15dB. d3) Proposed, PSNR =28.25dB.

Figure 4. Coefficient quantization for Case 2.

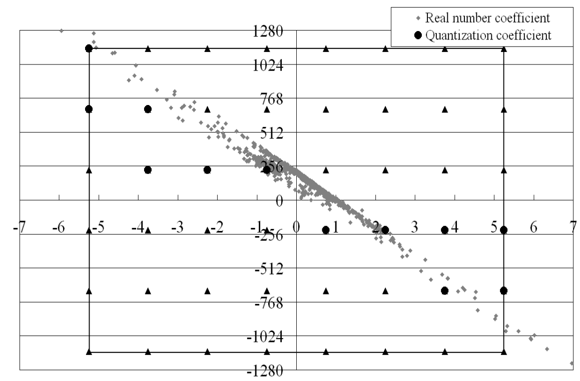


Figure 5. Distribution of p, q coefficients for F16, linear quantization method with 3bits for each coefficient.

Tables 2 and 3 show the detailed results for Case 1 and 2, respectively. In the tables, the first number in each cell is the PSNR in dB. The second number is the amount of levels occupied and the third is the corresponding entropy. To illustrate the source of improvement in the proposed method, consider, for instance, the first column in Table 2, where 2bits are used for each of p and q, resulting in a total of 2^4=16 quantized levels for the coefficients. The rows of Lena image in Table 2 show that the traditional linear method uses only 5 out of the 16 levels, while the proposed method uses all of the 16 levels. The entropy of the proposed method is 3.58, which is close to the theoretical value of 4. But it is only 1.87 for the linear method. Since the proposed method provides finer quantization steps and makes use of more quantization levels, better image quality can be obtained, which is 26.76dB while the linear method only yields 25.17dB.

Similar results for Case 2 are shown in Table 3. For Lena, the occupied levels are 12 and 56 out of 64 and the entropy values are 2.06 and 4.58 for linear and the proposed methods, respectively. Since much more levels are occupied, the proposed method yields quality of 27.04dB, which is much better than 22.87dB of the linear method. These tables also show that it is possible for our method to use fewer bits for quantization to achieve comparable image quality obtainable by the

Table 3. Experimental results for Case 2 (PSNR, occupied levels, and entropy).

Bits for *p*, *q*		2, 2	2, 3	3, 3	3, 4	3, 7	4, 8	5, 7
Quantized Levels		16	32	64	256	1024	4096	4096
Lena	Linear	7.67	7.17	22.87	20.74	26.50	27.67	27.94
		4	6	12	23	71	168	155
		1.91	1.00	2.06	3.03	4.56	6.35	6.19
	Proposed	7.54	26.59	27.04	27.46	28.23	28.60	28.71
		16	29	56	86	2498	505	511
		3.01	3.53	4.58	5.20	7.32	8.63	8.66
Baboon	Linear	11.81	10.78	17.92	16.50	18.74	19.76	19.82
		4	5	7	10	48	144	124
		1.03	0.69	0.55	2.91	3.89	6.16	5.79
	Proposed	18.92	19.32	19.54	19.70	19.93	20.01	20.10
		9	16	30	53	239	519	510
		2.41	2.89	3.66	4.68	7.17	8.68	8.65
F16	Linear	9.88	8.61	10.41	13.20	23.28	25.92	25.73
		4	7	8	13	59	136	128
		0.42	2.08	2.08	2.46	3.70	5.42	5.36
	Proposed	22.97	22.73	25.71	24.84	26.80	26.34	24.75
		8	13	18	31	152	335	329
		1.22	2.52	2.77	3.60	6.18	7.67	7.62
Pepper	Linear	10.34	9.43	19.15	23.69	27.38	28.85	29.10
		4	6	13	23	85	163	168
		1.62	1.35	1.72	3.45	4.73	6.31	6.27
	Proposed	25.15	27.17	28.25	28.61	29.53	30.06	30.11
		15	26	47	72	268	524	537
		2.92	3.44	4.2	5.10	7.47	8.71	8.75

linear method. For example, in Table 3, linear method uses 3 and 7bits to obtain an image quality of 26.50dB, whereas the proposed method uses only 2 and 3bits to obtain quality of 26.59dB. This translates to a reduction of 5bits to represent a range block in the fractal code.

6. Conclusions

In this paper, a new coefficient quantization method for fractal image compression is proposed. In the encoding phase, a pair of translation matrix and axis rotation matrix are used to transform p and q coefficients to p' and q'. Then we perform linear quantization in the transformed space and inverse-transform back to the original space. This method makes better use of the quantization levels according to the distribution of the coefficients p and q. Therefore, in comparison to the traditional method, this method can improve image quality when the same amount of quantization levels is used. On the other hand, to obtain the same image quality, fewer levels are required for the proposed method to achieve a better compression ratio. Experiments show the proposed method outperforms the SA optimization method computationally and compares favorably against the traditional linear coefficient quantization method, especially when fewer bits are allocated for the coefficients.

References

[1] Barnsley M. and Demko S., "Iterated Function System and the Global Construction of Fractals," *in Proceedings of Royal Society*, London, vol. A399, pp. 243-275, 1985.

[2] Chen H., Chung K., and Hung J., "Novel Fractal Image Encoding Algorithm Using Normalized One-Norm and Kick-out Condition," *Image and Vision Computing*, vol. 28, no. 3, pp. 518-525, 2010.

[3] Duh D., Jeng J., and Chen S., "Direct Allocating the Dihedral Transformation for Fractal Image Compression," *Journal of Information Science and Engineering*, vol. 23, no. 2, pp. 629-640, 2007.

[4] Fisher Y., *Fractal Image Compression: Theory and Application*, Springer-Verlag, New York, 1994.

[5] Ghazel M., Khandani A., and Vrscay E., "Vrscay, Improving Fractal Image Compression Schemes Yhrough Quantization and Entropy Coding," *in Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering*, Waterloo, pp. 661-664, 1998.

[6] Hamad N., "Hiding Text Information in a Digital Image Based on Entropy Function," *The International Arab Journal of Information Technology*, vol. 7, no. 2, pp. 146-151, 2010.

[7] Hsieh F., Wang C., Lee C., and Fan K., "A Lossless Image Coder Integrating Predictors and Block-Adaptive Prediction," *Journal of Information Science and Engineering*, vol. 24, no. 5, pp. 1579-1591, 2008.

[8] Hussain S., Qureshi K., Al-Mullah M., and Rashid H., "Local Predecimation with Range Index Communication Parallelization Strategy for Fractal Image Compression on a Cluster of Workstations," *The International Arab Journal of Information Technology*, vol. 6, no. 3, pp. 293-296,

2009.
[9] Jeng J., Tseng C., and Hsieh J., "Study on Huber Fractal Image Compression," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 995-1003, 2009.
[10] Li J., Sun C., and Grzybowski S., "Partial Discharge Image Recognition Influenced by Fractal Image Compression," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 15, no. 2, pp. 496-504, 2008.
[11] Mandelbrot B., *The Fractal Geometry of Nature*, Freeman and Company, New York, 1983.
[12] Oien G., "Parameter Quantization in Fractal Image Coding," *in Proceedings of IEEE International Image Processing*, pp. 142-146, 1994.
[13] Palazzari P., Coli M., and Lulli G., "Massively Parallel Processing Approach to Fractal Image Compression with Near-Optimal Coefficient Quantization," *Journal of Systems Architecture*, vol. 45, no. 10, pp. 765-779, 1999.
[14] Peitgen H., Henriques J., and Penedo L., *Fractals in the Fundamental and Applied Sciences*, Elsevier Science Publishing Company Inc., New York, 1991.
[15] Sankaragomathi B., Ganesan L., and Arumugam S., "Encoding Video Sequences in Fractal-Based Compression," *Fractals-Complex Geometry Patterns and Scaling in Nature and Society*, vol. 15, no. 4, pp. 365-378, 2007.
[16] Shannon C., "Prediction and Entropy of Printed English," *Bell Systems Technical Journal*, vol. 30, pp. 50-64, 1951.
[17] Takezawa M., Hond H., Miura J., Haseyama H., and Kitajima H., "A Genetic-Algorithm Based Quantization Method for Fractal Image Coding," *in Proceedings of International Conference on Image Processing*, Kobe, pp. 458-461, 1999.
[18] Wang X. and Chen Z., "A Fast Fractal Coding in Application of Image Retrieval," *Journal of Fractals-Complex Geometry Patterns and Scaling in Nature and Society*, vol. 17, no. 4, pp. 441-450, 2009.
[19] Wang X. and Wang S., "An Improved No-Search Fractal Image Coding Method Based on a Modified Gray-Level Transform," *Computers and Graphics*, vol. 32, no. 4, pp. 445-450, 2008.
[20] Wu M., Teng W., Jeng J., and Hsieh J., "Spatial Correlation Genetic Algorithm for Fractal Image Compression," *Chaos, Solitons and Fractals*, vol. 28, no. 2, pp. 497-510, 2006.

Jyh-Horng Jeng received his BS and MS degree in mathematics from Tamkang University, Taiwan in 1982 and 1984, respectively, and the PhD degree in mathematics from State University of New York at Buffalo in 1996. He was a senior research engineer with Chung Shan Institute of Science and Technology, Taiwan from 1984 to 1992. Currently, he is a professor of Department of Information Engineering, I-Shou University, Taiwan. His research interests include multimedia applications, soft computing and machine learning, error correcting code, and VLSI architecture design.

Shuo-Li Hsu received his BS degree and the MS degree in information engineering from the I-Shou University, Taiwan, in 2001 and 2003, respectively. He is currently, working toward the PhD degree in information engineering at I-Shou University, Taiwan. His research interests include soft computing and machine learning, image processing, and multimedia networks.

Yukon Chang received his PhD degree in computer science from Pennsylvania State University, University Park, in 1986. He was an assistant professor of computer science with the State University of New York at Albany from 1986 to 1992. In 1992, he joined the Department of Information Engineering, I-Shou University, Kaohsiung, Taiwan, where he is now an associate professor. He also served as director or Computer Center at the I-Shou University from 1993 to 1996, Director of the Library from 1998 to 2001, and Chair of the Department of Information Engineering from 2002 to 2005. His primary research interest is in multimedia networks.