An improved fast fractal image compression using spatial texture correlation*

Wang Xing-Yuan(王兴元)†, Wang Yuan-Xing(王远星), and Yun Jiao-Jiao(云娇娇)

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, *China*

(Received 2 March 2011; revised manuscript received 6 May 2011)

This paper utilizes a spatial texture correlation and the intelligent classification algorithm (ICA) search strategy to speed up the encoding process and improve the bit rate for fractal image compression. Texture features is one of the most important properties for the representation of an image. Entropy and maximum entry from co-occurrence matrices are used for representing texture features in an image. For a range block, concerned domain blocks of neighbouring range blocks with similar texture features can be searched. In addition, domain blocks with similar texture features are searched in the ICA search process. Experiments show that in comparison with some typical methods, the proposed algorithm significantly speeds up the encoding process and achieves a higher compression ratio, with a slight diminution in the quality of the reconstructed image; in comparison with a spatial correlation scheme, the proposed scheme spends much less encoding time while the compression ratio and the quality of the reconstructed image are almost the same.

Keywords: fractal image compression, texture features, intelligent classification algorithm, spatial correlation

PACS: 42.30.Va, 42.30.Wb, 87.57.C– **DOI:** 10.1088/1674-1056/20/10/104202

1. Introduction

Fractal image compression was originally proposed by Barnsley *et al.*[1−3] and first realized by Jacquin.[4] The underlying premise of fractal image compression is based on the partitioned iteration function system (PIFS) which utilizes the self-similarity property in the image to achieve the purpose of compression.

To encode an image according to the self-similarity property, each range block must find the most similar domain block in a large domain pool. For a baseline method, the encoding process is time consuming since a large number of computations of similarity measurement are required to find the best match. Also, in order to achieve global optimization, global offsets have to be recorded, which increases the storage space required. Therefore, the aims of fractal image compression are to speed up the encoder and to increase the compression ratio.

The bottleneck in the PIFS fractal coding scheme is the time spent in the encoding process. In order to alleviate this serious encoding time problem, several efficient fractal encoding algorithms have been developed. These encoding algorithms include the domain pool selection approach,[5] the partitioned-based approach,[6−11] and the search strategy-based approach.[12,13] Truong *et al.*[14] presented an efficient spatial-correlation-based algorithm for fractal encoding and their proposed algorithm had a significant improvement in comparison with some typical image compression algorithms.

In this paper, we propose an improved fractal image compression scheme by using spatial texture correlation, which makes full use of correlation between blocks. The experimental results show that this scheme can significantly speed up the encoding process and achieve a high compression ratio, and the quality of the reconstructed image is almost the same as that with the baseline scheme.

The organization of this paper is as follows. In section 2 a brief review of the baseline fractal image compression method is described. Section 3 analyses the spatial correlation algorithm and discusses its drawbacks. In section 4 we give a spatial texture correlation fractal image compression scheme to improve the algorithm. Experimental results including encoding time, compression ratio, and peak signal-to-noise

*Project supported by the National Natural Science Foundation of China (Grant Nos. 60573172 and 60973152), the Superior University Doctor Subject Special Scientific Research Foundation of China (Grant No. 20070141014), and the Natural Science Foundation of Liaoning Province of China (Grant No. 20082165).

†Author to whom any correspondence should be addressed. E-mail: wangxy@dlut.edu.cn

ratio (PSNR) are given in section 5. Section 6 presents the conclusions.

2. Baseline fractal image compression

The baseline fractal image compression scheme (BFC) is based on contractive transformations and PIFS in a two-dimensional metric space. First, the original image is divided into non-overlapping square blocks of size $B \times B$ (denoted range blocks). These range blocks cover the entire image. Overlapping domain blocks of size $C \times C$ ($C > B$ to ensure contractivity of PIFS, with a common value of $C = 2B$) are constructed from the original image by sliding a window of size $C \times C$ over the entire image with a fixed step size β. Usually we choose the same step size β in the horizontal and the vertical direction. Selecting β as a power of 2 to make it easy to generate the domain pool. For this research, $\beta = 2$.

As domain blocks are larger than range blocks, each domain block is spatially contracted before the range block matching process. Pixel averaging can be used to transform the $C \times C$ size domain block into a $B \times B$ size range block, and these contracted domain blocks are denoted as the domain block pool Ω.

After the range blocks and domain block pool are created, for each range block R we search the domain pool to find the best matching domain block D and its corresponding transformation φ, this $\varphi(D)$ provides the minimum value in the least square sense between range block R and domain block D. The transformation φ consists of the composition of a scaling s and a luminance offset o, as $\varphi(D) = sD + oI$. In other words, we want to minimize the following matching error function:

$$\begin{aligned} E(R,D) &= \|sD + oI - R\|^2 \\ &= \sum_{i=1}^{B}\sum_{j=1}^{B}(sd_{ij} + oI - r_{ij})^2, \end{aligned} \tag{1}$$

where r_{ij} and d_{ij} are individual pixels in the range block and the domain block, respectively. I is a block of the same size as R, but with all elements equal to 1. A small error value implies that the range block is close to the transformed domain block; otherwise the range block and domain block are not considered similar. With respect to the parameters s and o, the optimal affine parameters can be obtained by the least squares method as follows:

$$s = \frac{\langle R - \bar{r}I, D - \bar{d}I\rangle}{\|D - \bar{d}I\|^2}, \quad o = \bar{r} - s\bar{d}, \tag{2}$$

where $\bar{r}$ and $\bar{d}$ are the mean value of the range block R and the domain block D, respectively. $\langle\cdot,\cdot\rangle$ is the inner product, and $\|\cdot\|$ is the usual two-norm.

In practice, the fractal encoding scheme based on Eq. (2) is not used since we need to constrain the range of the scaling parameter and quantize the parameter for compression. A set of quantized fractal parameters $\{s_i\}$ and $\{o_k\}$ are used to obtain the minimum of

$$E(R,D) = \min_i \min_k \|R - s_i D - o_k I\|^2 . \tag{3}$$

Parameters i and k are determined by the bit allocations of s and o. Based on Eq. (2), we will otain the minimum matching error function

$$\begin{aligned} E(R,D) &= \min_i \|R - s_i D - (\bar{r} - s_i\bar{d})I\|^2 \\ &= \min_i \|R - \bar{r}I - s_i(D - \bar{d}I)\|^2 . \end{aligned} \tag{4}$$

Therefore, the range block R should have the following relationship with its best matched domain block D:

$$R \approx s_i D + o_k I = s_i(D - \bar{d}I) + \bar{r}I. \tag{5}$$

In this expression, the algorithm parameters are the range block's mean value $\bar{r}$, and the scaling coefficient s_i, $\bar{d}$ can be obtained from the reconstructed image. Based on the contractive mapping fixed point theorem and the collage theorem,[11] the reconstructed image will converge to an approximation of the original image quickly.

One advantage of this method is that the mean value of the range block is a positive value and less than the maximum pixel value in the image. Another advantage is that the correlation between s and $\bar{r}$ is sufficiently small so that it can be ignored, and this makes the method more efficient than the original one.

3. Analysis of the spatial correlation fractal image compression

3.1. Review of the spatial correlation scheme

The primary disadvantage of BFC is the large amount of time required to search for the best range-domain block match from the domain pool. Many

algorithms have been proposed to address this problem. According to the spatial correlation between the current range block and the four neighbouring range blocks, Truong *et al.*[14] presented an efficient fractal encoding algorithm. As shown in Fig. 1, the current range block is denoted by $R_{\rm c}$ and the four neighbouring range blocks are denoted by $R_{\rm nw}$, $R_{\rm n}$, $R_{\rm ne}$, and $R_{\rm w}$, respectively, to the northwest, north, northeast and west. The best matched domain blocks of $R_{\rm nw}$, $R_{\rm n}$, $R_{\rm ne}$, and $R_{\rm w}$ are denoted by $D_{\rm nw}$, $D_{\rm n}$, $D_{\rm ne}$, and $D_{\rm w}$, respectively, in the domain pool.

In Ref. [14], for the current range block $R_{\rm c}$, 16 domain blocks should be examined. Among these 16 examined domain blocks, each set consisting of four consecutive domain blocks is considered for each concerning direction. For example, for a west direction of current range block $R_{\rm c}$, four domain blocks, say $D^0_{\rm w}$, $D^1_{\rm w} = (D_{\rm w})$, $D^2_{\rm w}$ and $D^3_{\rm w}$ should be examined. Truong *et al.*'s search strategy is, first, to try to find the best matched contracted domain block of $R_{\rm c}$ under fractal affine transform among the 16 concerned domain blocks. If the collage error between the corresponding best matched contracted domain block and the current range block $R_{\rm c}$ is less than the threshold, only 4 bits are required to record the offset of the domain block (2 bits are used to record the range correlation and the other two to record the domain correlation as depicted in Fig. 1), in addition, the fractal encoder should record the scaling coefficient s_i and the range block's mean value $\bar{r}$. Otherwise, the full search strategy is employed to find the best matched domain block in the domain pool.

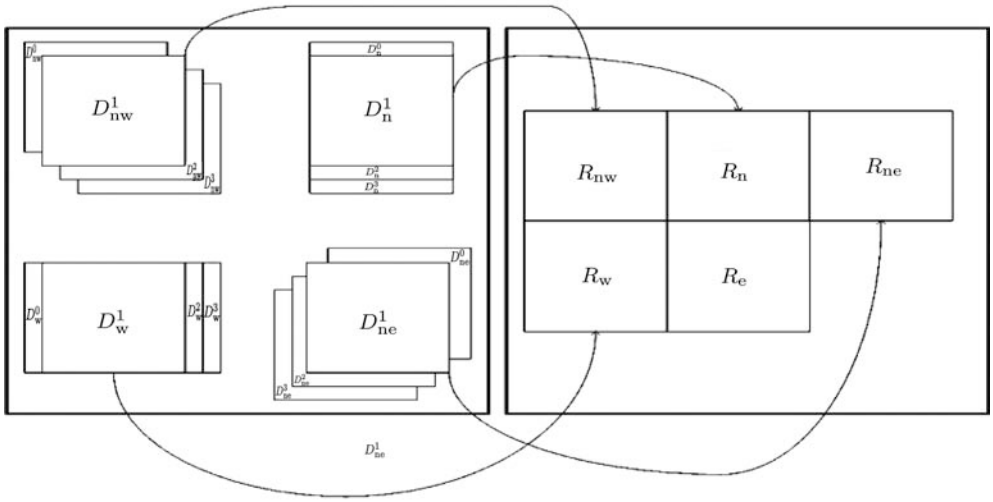


Fig. 1. The searching space of the current range block in the spatial correlation approach.

3.2. The drawbacks and the solution

The spatial correlation reveals that neighbouring blocks usually have some similar properties such as edge and shade, etc. Moreover, the characteristics of the spatial correlation depend on the orientations of the edge and shade. For instance, consider the eight nearest neighbouring blocks of a given block. If the given block possesses a horizontal edge, its left and right neighbours usually possess similar horizontal edges, but not the other six neighbours. Similarly, if the block possesses a diagonal edge, then its left-up and right-down neighbours usually possess a diagonal edge, but not the others. Based on this property, one can limit the searching space of the current block to the matched domain blocks of the neighbouring range blocks. Since the searching space is much smaller than that of the full search method, the compression speed is improved. On the other hand, in order to avoid poor matches using this mechanism, one also pre-defines a threshold to determine if a full search process for this range block should be invoked. Thus the quality of the retrieved image can be maintained.

However, in contrast with the mechanism of spatial correlation, for a range block, it first finds the best matched domain block among the 16 concerned domain blocks of its four neighbouring range blocks no matter whether or not its four neighbouring range blocks have similar properties to the current range block. The following possible full search reveals that the matching error computations with these 16 domain blocks are repeated; in addition, even if the

current range block finds an acceptable domain block among the 16 blocks, it does not mean that all of its four neighbouring range blocks have similar properties to it, thus some matching error computations with the concerned domain blocks of the dissimilar range block are also not necessary. Another drawback of the spatial correlation scheme is that instead of a full search, a set of most likely matched domain blocks should be searched for a certain range block in order to improve the search speed, which corresponds to the mechanism of correlation.

Self-similarity is one important property of fractals. For real-world images that have a local self-similarity, fractal image coding finds the most similar domain block of the range block. It means that the matching of two blocks depends on some measure of their similarity. Texture features is an important property of an image, and it is usually utilized for image retrieval applications.[15−23] The spatial correlation reveals that neighbouring blocks usually have similar texture features. In order to avoid doing any unnecessary and repeated matching error computations and to make full use of correlation, we first calculate the similarity between the current range block $R_{\rm c}$ and each of its four neighbouring range blocks. If the similarity between $R_{\rm c}$ and neighbouring range block R_X (X=nw, n, ne, w) is less than one threshold δ, then we find the best matched domain block among the four domain blocks of block R_X; otherwise the four domain blocks of R_X should not be examined. If the similarity between $R_{\rm c}$ and its four neighbouring range blocks are all greater than the threshold, a set of the most similar matched domain blocks will be searched for the current range block $R_{\rm c}$ in order to further improve the search.

4. Texture correlation fractal image compression

4.1. Texture features and similarity

The co-occurrence matrix[20−23] is a two-dimensional histogram that estimates the pair-wise statistics of gray levels. The (i, j)-th element of the co-occurrence matrix represents the estimated probability that gray level i co-occurs with gray level j at a specified displacement d and angle θ, $i = d\cos\theta$ and $j = d\sin\theta$. By choosing the values of d and θ, a separate co-occurrence matrix is obtained. From each co-occurrence matrix a number of textural features can be extracted. To measure the spatial similarity, we use two features: entropy and maximum entry along with associated direction. More features can be included for better accuracy, but it will be at the cost of volume of data storage and search speed.

Obtain the co-occurrence matrix for four (horizontal 0°, vertical 90°, and two diagonal 45° and 135°) orientations in the block and normalize the entries of four matrices to [0, 1] by dividing each entry by the total sum of the matrix. Then extract the average entropy value from the four matrices, and the largest value among the entries of four matrices along with each associated direction through the following equations.

Entropy:

$$e = -\sum_i \sum_j p(i,j)\log(p(i,j)). \tag{6}$$

Maximum entry:

$$p = \max(p(i,j)), \tag{7}$$

where $p(i, j)$ is the (i, j)-th entry of the normalized co-occurrence matrix.

Let e_q and p_q be the entropy and maximum entry for block q in the image, the texture feature (TF) of block q is defined as

$$\mathrm{TF}_q = w_e \cdot e_q + w_q \cdot p_q, \tag{8}$$

where w_e and w_p are weights for each e, p which are subjected to $w_e + w_p + \varepsilon = 1$ (ε is a small enough value except weights for each e, p). We set w_e= 0.7, $w_p \doteq$0.3 because entropy has more power than maximum entry to discriminate texture.[24] The approximate expression of Eq. (8) is

$$\mathrm{TF}_q = w \cdot e_q + (1-w) \cdot p_q. \tag{9}$$

Recently, Xiao *et al.*[25] proposed a new algorithm for image similarity measure which makes use of an edge direction histogram (EDH) to characterize the structure of the graph, and estimates the dissimilarity of graphs by computing the distance of EDHs without defining cost functions. At the same time, Torre *et al.*[26] conduct a complete metric space (Y, d_Y) of measure-valued images, $\mu : X \to \mathbf{M}(\mathbb{R}_g)$, where X is the base or pixel space and $\mathbf{M}(\mathbb{R}_g)$ is the set of probability measures supported on the greyscale range $\mathbb{R}_g$ ($\mathbb{R}_g \subset \mathbf{R}, \mathbb{R}_g = [0, 1]$). Then the space (Y, d_Y) is employed with a general model of affine self-similarity of images. It can also measure the block similarity by the following equation:

$$d(Q,I)=\sqrt{(R_0-R'_0)^2+(R_1-R'_1)^2+(R_2-R'_2)^2+\cdots+(R_{2m\times n-1}-R'_{2m\times n-1})^2},$$

where Q and I are spatial distribution features which are defined as follows:

$$Q=(R_0,R_1,R_2,\ldots,R_{2m\times n-1}),$$
$$I=(R'_0,R'_1,R'_2,\ldots,R'_{2m\times n-1}).$$

That is the distance between texture elements and its centre of mass in a spatial distribution map. Due to the difficulty of reasonably defining edit operations in graph edit distance and affine self-similarity of images, we used another method to measure block similarity which also follows the law of measuring the distance of texture feature distributions between two images or blocks.

The block similarity between block q_1 and q_2 is then given by the following equation:

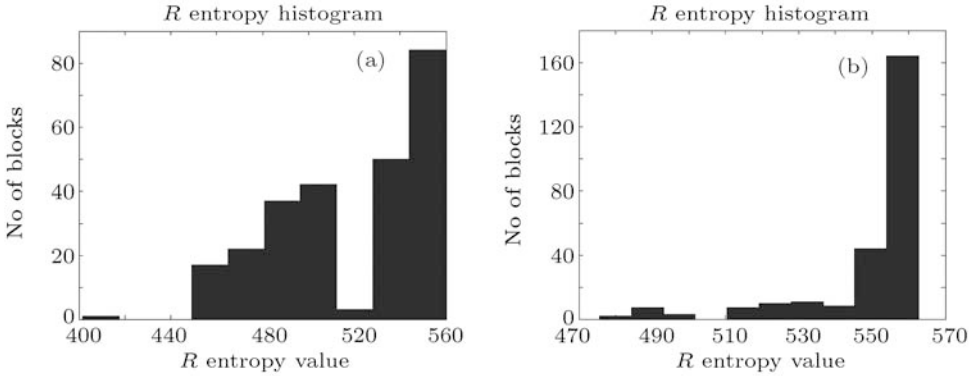
$$S(q_1,q_2)=|\mathrm{TF}_{q_1}-\mathrm{TF}_{q_2}|\,. \tag{10}$$

If the similarity $S(q_1,q_2)$ is less than the threshold δ, we will say that block q_1 and q_2 have similar texture features.

Fig. 2. A pair of blocks of the Lena image for which S is large and another is small: (a) a pair of blocks whose S is small. $S=0.2482$, $\mathrm{TF}_{q_1}=484.2925$, $\mathrm{TF}_{q_2}=485.0443$; (b) a pair of blocks whose S is large. $S=148.3974$, $\mathrm{TF}_{q_1}=550.1751$, $\mathrm{TF}_{q_2}=401.7777$.

In Fig. 2, we demonstrate the effectiveness of their block similarity $S(q_1,q_2)$ as defined in Eq. (10). Here we show a pair of blocks of the Lena image whose S is small and another pair of blocks whose S is large. Four parameters under the image block are the x and y positions of the left and top coordinates, and the width and the height of the image block.

As an illustrative example, figure 3 shows the results of the statistics histogram of texture features and the similarity of the Lena and Baboon images with size 256×256. For convenience, we coverted all values to integers by multiplication. Figures 3(a) and 3(b) show the entropy histogram of R blocks, figures 3(c) and 3(d) show the entropy histogram of D blocks. Figures 3(e) and 3(f) show the similarity histogram between blocks R and D. From the statistics figures 3(a)–(d), we can see that the greater the number of blocks, the greater is the block similarity. At first glance, it would appear that the two images are quite translationally self-similar since there are peaks in both distributions in the interval [550, 560] (Figs. 3(a) and 3(b)). The peaks of two other distributions appear in [650, 660] and [690, 700]. Taken together, the Baboon image is more pronounced. As the last two statistics histograms shown in Figs. 3(e) and 3(f) could present histogram distributions of the similarities $S(q_1,q_2)$ for all possible pairings, but whose range blocks can be viewed as the most likely matching domain blocks, we use an intelligent classification algorithm for further image processing.



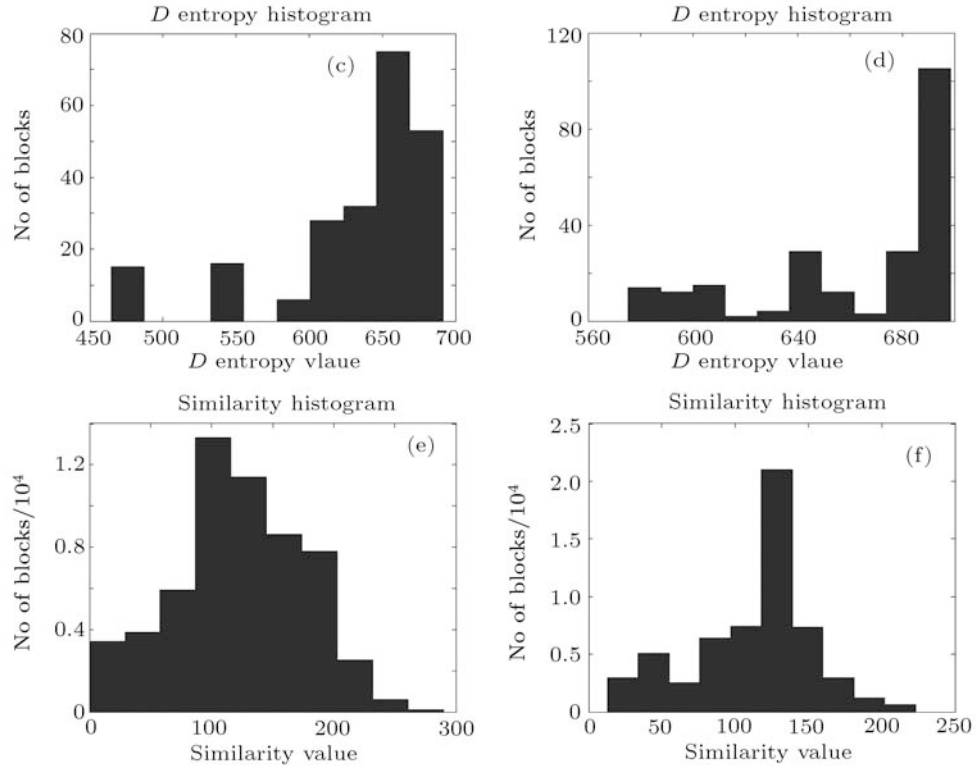


Fig. 3. The statistics histogram of texture features and the similarity of the Lena and Baboon images with size 256×256. Panels (a) and (b) are the entropy histograms of R blocks; panels (c) and (d) are the entropy histograms of D blocks; panels (e) and (f) are the similarity histograms between blocks R and D. (a) The R entropy histogram of the Lena image. (b) The R entropy histogram of the Baboon image. (c) The D entropy histogram of the Lena image. (d) The D entropy histogram of the Baboon image. (e) A similarity histogram between R and D of the Lena image. (f) A similarity histogram between R and D of the Baboon image.

4.2. Intelligent classification algorithm (ICA)

The traditional fractal domain block search process is linear with respect to the domain block location in the domain pool. The advantage of this method is that implementation is simple. However, a significant disadvantage is that for each range block all of the domain blocks will be processed and result in many, ultimately unnecessary, failed range-domain block comparisons. Significant time is wasted using this basic search process. The intelligent classification algorithm (ICA)[27] approach can resolve this problem. The ICA functions by grouping all domain blocks that have a similar texture feature value together, and constructing a location index vector consisting of positional information for all domain blocks in all defined groups. Based on the range block TF value, the algorithm searches only those domain blocks the have a similar TF and are therefore logically grouped together. A more detailed background is given in Ref. [27].

The algorithm begins by determining the maximum of the rounded domain block TF values. This maximum value, determines the length of a TF number vector. Elements in the TF number vector store the number of domain blocks having this TF value. A TF index vector is derived from the TF number vector and is used to store the beginning position for this TF value in a location vector. After the TF index vector is created, positional information for all domain blocks is placed in the location vector based on the previously determined domain block TF values. Data in the location vector therefore consists of domain location information sorted by TF value. Note that usually the value TF of a block is within a small range, with the aim to effectively utilize ICA, the TF value in Eq. (8) should be made a little change by multiplying 100, thus the maximum of the rounded domain block TF values will be large enough to group effectively.

4.3. Texture correlation scheme

The selection of an appropriate similarity threshold δ between two blocks is very important. If this

threshold is large, it is easy for range block $R_{\rm c}$ to have a texture correlation with its four neighbouring range blocks, so fewer domain blocks will be searched and a higher bit rate will be obtained. If the threshold is small, more range blocks will find their best matched domain blocks by using ICA, thus longer encoding times are therefore required and a lower bit rate will be obtained. Determining an appropriate threshold to balance the encoding time and bit rate is critical.

To avoid large gaps between this local minimum obtained through the best matched domain block of neighbouring range blocks and the global minimum obtained through the ICA method, a similar threshold T is pre-defined.[14] If the local minimum exceeds this threshold, the ICA search will be invoked. The ICA search method belongs to a class of optimal domain block searching algorithms that only search a set of most likely matching domain blocks for a certain range block, thus improving the search and corresponding encoding speed. Different from an adaptively changing similarity threshold,[27] in our work, the domain blocks whose TF value satisfies $S(q_R, q_D) \leq \lambda$ can be viewed as the most likely matching domain blocks, and the threshold λ is a certain value.

Combining all the methods previously described, we give the detailed steps of our improved texture correlation scheme as follows.

(i) Duplicate the original image, one image is partitioned into non-overlapping range blocks with size $B \times B$, and another is partitioned into overlapping domain blocks with size $2B \times 2B$.

(ii) Calculate the TF value for each range block.

(iii) Contract each domain block to the size of the range block, calculate the TF value for each contracted domain block and classify contracted domain blocks using ICA.

(iv) For each range block R_i located at the first row and column, the ICA search strategy is employed to find the best matched domain block using Eq. (4) in the domain pool. Suppose the TF value of R_i is TF_i, then find its best matched contracted domain block among the domain blocks with TF values in the range of $[\mathrm{TF}_i - \lambda, \mathrm{TF}_i + \lambda]$, and store the fractal transformation coefficients.

(v) For each range block $R_{\rm c}$ not located at the first row or column, define its searching space $S_{\rm c}$ which is empty. Suppose the TF value of $R_{\rm c}$ is $\mathrm{TF}_{\rm c}$ and the texture feature value of its neighbouring range block R_X ($X = \mathrm{nw}, \mathrm{n}, \mathrm{ne}, \mathrm{w}$) is TF_X, if $|\mathrm{TF}_{\rm c} - \mathrm{TF}_X| \leq \delta$, add four consecutive domain blocks corresponding to range block R_X into the searching space $S_{\rm c}$. If $S_{\rm c}$ is not empty, find its best matched contracted domain block $D_{\rm c}$ from $S_{\rm c}$, if the collage error between $D_{\rm c}$ and $R_{\rm c}$ is less than the threshold T, store the correlation fractal transformation coefficients. If $S_{\rm c}$ is empty or the collage error between $D_{\rm c}$ and $R_{\rm c}$ is not less than the threshold T, find its best matched domain block among the domain blocks with TF values in the range $[\mathrm{TF}_i - \lambda, \mathrm{TF}_i + \lambda]$ using ICA and store the fractal transformation coefficients.

In step (ii), the TF value of each range block is calculated in the pre-process phase, in this way, the calculations of TF values of the current range block $R_{\rm c}$ and its neighbouring range blocks will be avoided and repeated in step (v). In step (iv), since the ICA search strategy is performed, the absolute position is stored. In step (v), if the searching space $S_{\rm c}$ of current range block $R_{\rm c}$ is not empty and the minimum collage error between $D_{\rm c}$ and $R_{\rm c}$ is less than T, the range block $R_{\rm c}$ is called a "hit" block, it stands for the current range block $R_{\rm c}$ having similar TF with its neighbouring range blocks. For such a hit block, only 4 bits are required to record the offset of the domain block instead of the absolute position. Two bits are used to record the range correlation and the other two to record the domain correlation as depicted in Fig. 1. Let $N_{\rm R}$ and $N_{\rm H}$ denote the number of range blocks and hit blocks, respectively. For hit blocks, 4 bits are required to record the relative positions; for non-hit range blocks ($N_{\rm R} - N_{\rm H}$ in total), the $B_{\rm A}$ bits are required to record the absolute positions. Let B_s denote the number of bits to represent the contrast coefficient. Then the bit rate [bit per pixel (bpp)] can be computed directly in terms of the number of hit blocks as

$$\mathrm{bpp} = \frac{N_{\rm H}(1 + 4 + B_s + 8) + (N_{\rm R} - N_{\rm H})(1 + B_{\rm A} + B_s + 8)}{N_{\rm TP}}, \tag{11}$$

where $N_{\rm TP}$ is the total number of pixels in the image. Note that 1 bit is required to indicate if the range block is a hit block or not, $\bar{r}$ is encoded using 8 bits.

5. Experimental results

The 256×256 images of Lena and Baboon, which are quite diverse in content and therefore are good for evaluating the performance of different image features, with a fixed 8×8 range block size are tested to demonstrate the encoding speed-up and quality of the proposed algorithm in comparison with the baseline method and spatial correlation scheme.[14] The software simulation is done using MATLAB7.0 on an AMD/Athlon 2.0 GHz Windows XP PC.

In our work, 2 bits are allocated for the scaling coefficient, 7 + 7 = 14 bits are used to represent the domain block location information. That is to say, $B_{\rm A} = 14$ and $B_s = 2$. In addition, the displacement d is set to be 1 to calculate the TF value. To compare our proposed texture correlation (TC) method with some other methods, we use speed-up rate, bpp and PSNR as the criteria of comparison. The definition of PSNR is

$$\mathrm{PSNR} = 10\log_{10}\left(255^2 \Big/ \left((1/N)\sum_{i=1}^{N}(f_i - g_i)^2\right)\right), \quad (12)$$

where f_i is the pixel of the original image and g_i is the pixel of the reconstructed image. N denotes the total number of pixels.

Table 1. The comparison of the baseline method, spatial correlation (SC) method, and our proposed texture correlation (TC) method with different threshold δ and fixed threshold $\lambda = 5$ and $T = 100$.

Image	Method	PSNR/dB	Time/s	bpp	Hit blocks	Speed-up rate
Lena	baseline	28.0342	1282.7	0.3750	0	1
	SC	27.6975	587.3750	0.3052	560	2.18
	TC with $\delta = 10$	27.2945	172.14	0.3253	428	7.45
	TC with $\delta = 20$	27.2832	168.75	0.3120	514	7.60
Baboon	baseline	21.1285	1282.7	0.3750	0	1
	SC	21.0951	1116.8	0.3696	138	1.15
	TC with $\delta = 10$	21.0789	620.7030	0.3735	112	2.07
	TC with $\delta = 20$	21.0723	608.6250	0.3712	127	2.11

Fig. 4. The results of the proposed method in comparison with the baseline method and the spatial correlation method: (a) original Lena image size 256×256; (b) baseline method, time used is 1282.7 s, PSNR = 28.0342, bpp = 0.3750; (c) spatial correlation method, time used is 587.3750 s, PSNR = 27.6975, bpp = 0.3052; (d) texture correlation method, time used is 168.75 s, PSNR = 27.2832, bpp = 0.3120; (e) the JPEG2000 method, PSNR = 28.6, bpp = 0.4000; (f) the EZW method, PSNR = 26.1, bpp = 0.4000.

A comparison of our proposed texture correlation (TC) method with other methods such as the baseline method and spatial correlation (SC) method has been made in Table 1. The threshold T is 100, λ is 5 and the threshold δ is 10 and 20, respectively. The experimental results indicate that in comparison with some typical methods, both spatial correlation and our proposed texture correlation method can speed up the encoding time and improve the bit rate while there is only a little PSNR decay. The advantage of our proposed method is seen more clearly on the speed-up rate. For the Lena image, with more or less the same bpp and PSNR, the spatial correlation method is 2.18 times faster while our proposed texture correlation method is about 7.53 times faster than the baseline method. In order to test our proposed texture correlation method, different thresholds δ are utilized here. As the comparison indicates, higher value of thresholds δ produces more hit blocks and higher compression ratio, but the quality will decrease. If the threshold λ is bigger, more domain blocks will be searched thus the quality of the reconstructed image will be improved at the expense of a longer encoding time.

As an illustrative example, figures 4 and 5 show the results of the proposed method in comparison with the baseline method and the spatial correlation method. Figure 4(a) is the original image with size 256×256. Figures 4(b), 4(c), and 4(d) show the reconstructed images using the baseline method, the spatial correlation method and our texture correlation method, respectively. With the threshold being set as $\delta = 20$, $T = 100$, and $\lambda = 5$. In comparison with the baseline method, our method is 7.60 times faster and the bit rate is also improved while there is only 0.75 dB decay; in comparison with the spatial correlation method, our method is 3.48 times faster, the bit rate and PSNR are almost the same.

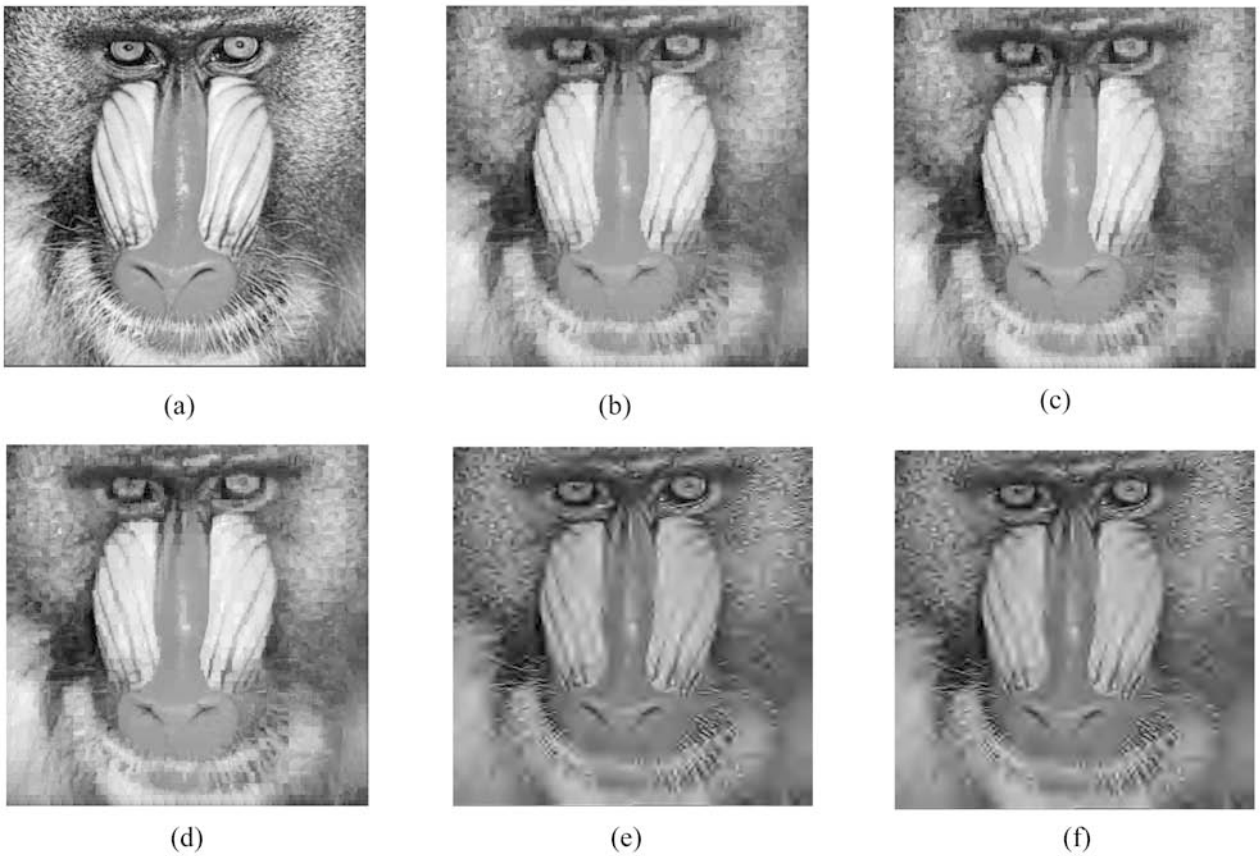


Fig. 5. The results of the proposed method in comparison with the baseline method and the spatial correlation method: (a) original Baboon image, size 256×256; (b) baseline method, time used is 1282.7 s, PSNR = 21.1285, bpp = 0.3750; (c) spatial correlation method, time used is 1116.80 s, PSNR = 21.0951, bpp = 0.3696; (d) texture correlation method, time used is 608.6250 s, PSNR = 21.0723, bpp = 0.3712; (e) the JPEG2000 method, PSNR = 22.1, bpp = 0.40004; (f) the EZW method, PSNR = 20.2, bpp = 0.4000.

Famous encoding algorithms called "No search" algorithms[28] do not search the domain pool, but a specific domain block is appointed to be the "best matching" one. An obvious advantage of the "No search" scheme is that there is no need to store the relative position vector. Therefore, better compression ratios can be achieved. Another new classification method is called AFD (approximated first derivative)–NRMS (normalized root mean square error).[29] The compression ratios are also significantly reduced by using AFD and NRMS parameters. A comparison of our proposed texture correlation method with other methods such as the no search method in Ref. [27], the AFD–NRMS method in Ref. [29], the JPEG2000 method and the EZW method have been made in Table 2. The threshold[30−33] δ is 10 and 20, respectively. In the region of image compression, bpp means bit per pixel, that is the bit number needed to store a pixel of image. So, the smaller the value of bpp, the better the performance. In our experimental results in Table 2, the value of PSNR is 27.2832 and bpp is 0.3120 using our texture correlation method while PSNR is 28.6 and bpp is 0.4000 using the JPEG2000 method for the Lena image. In the case PSNR was slightly low as compared to JPEG2000's, the bpp is smaller than the JPEG2000's, this fact indicates that our texture correlation algorithm has a higher compression ratio in comparison with JPEG2000. The experimental results indicate that in comparison with the no search method, the AFD–NRMS method, the JPEG2000 method,[34,35] and the EZW method,[36] our proposed texture correlation method can improve the bit rate greatly while there is only a little PSNR decay. For the Baboon image, with more or less the same PSNR, our proposed texture correlation method is about 1.08 times smaller than the JPEG2000 and EZW methods and 2.16 times smaller than the no search and AFD–NRMS methods.

Table 2. A comparison of the no search method, the AFD–NRMS method in Ref. [27], the JPEG2000 method, the EZW method and our proposed texture correlation method with different thresholds δ.

Image	Method	PSNR/dB	bpp
Lena	No search	25.84	0.8000
	Ref. [28]	32.00	0.8000
	JPEG2000	28.60	0.4000
	EZW	26.10	0.4000
	TC with $\delta = 10$	27.2945	0.3253
	TC with $\delta = 20$	27.2832	0.3120
Baboon	No search	20.46	0.8000
	Ref. [28]	22.37	0.8000
	JPEG2000	22.10	0.4000
	EZW	20.20	0.4000
	TC with $\delta = 10$	21.0789	0.3735
	TC with $\delta = 20$	21.0723	0.3712

6. Conclusion

In this paper, the spatial texture correlations in both the domain and range blocks and the ICA search strategy are utilized to speed up the encoding process and improve the bit rate for fractal image compression. In comparison with the spatial correlation method, the texture correlation between neighbouring blocks in an image is used in our proposed method. The experimental results indicate that texture correlation is one of the most important properties for the representation of an image. Entropy and maximum entry from co-occurrence matrices are used for representing texture correlation in the image. For a range block, concerned domain blocks of neighbouring range blocks with a similar texture correlation can be searched to find a local minimum, thus it has a lower computational complexity. With a view to avoiding large gaps between the local minimum and the global minimum obtained through the baseline method, threshold T is employed. In addition, domain blocks with a similar texture correlation are searched in the ICA search process to further make use of the texture correlation between blocks and speed up the encoding process. Experiments show that in comparison with some typical methods, our proposed algorithm significantly speeds up the encoding process and achieves a higher compression ratio, with a slight diminution in the quality of the reconstructed image; in comparison with the spatial correlation scheme, our proposed scheme uses much less encoding time while the compression ratio and the quality of the reconstructed image is almost the same.

References

[1] Barnsley M F and Demko S 1985 *Math. Phys. Sci.* **399** 243

[2] Barnsley M F 1988 *Fractal Everywhere* (New York: Academic) p. 34

[3] Barnsley M F, Elton J H and Hardin D P 1989 *Constr. Approx.* **5** 3

[4] Jacquin A E 1992 *IEEE T. Image Process.* **1** 18

[5] Barthel K U and Voye T J 1994 *Proc. Int. Workshop Image Process.* Budapest, Hungary

[6] Davoine F and Chassery J M 1994 *12th Int. Conference Pattern Recogn.* (Los Alamitos: IEEE Computer Society Press) **1** 801

[7] Davoine F, Svensson J and Chassery J M 1995 *Proc. Int. Conference Image Process.* Washington D.C. **3** 284

[8] Davoine F, Antonini M, Chassery J M and Barlaud M 1996 *IEEE T. Image Process.* **5** 338

[9] Fisher Y 1994 *Fractal Image Compression, Theory and Application* (New York: Springer-Verlag)

[10] Saupe D and Jacob S 1997 *Electron. Lett.* **33** 46

[11] Fisher Y 1995 *Fractal Image Compression: Theory and Applications* (Berlin: Springer-Verlag) p. 121

[12] Furao S and Hasegawa O 2004 *Signal Process. Image* **19** 393

[13] Kominek J 1995 *Proc. SPIE* **2419** 296

[14] Truong T K, Kung C M, Jeng J H and Hsieh M L 2004 *Chaos, Solitons and Fractals* **22** 1071

[15] Yoo H W, Jang D S, Jung S H, Park J H and Song K S 2002 *Pattern Recogn.* **35** 749

[16] Ma W Y and Manjunath B S 1996 *Proc. IEEE Int. Conference Comput. Vision Pattern Recog.* Boston, USA p. 425

[17] Manjunath B S and Ma W Y 1996 *IEEE TPAMI* **18** 837

[18] Tamura H, Mori S and Yamawaki T 1978 *IEEE Trans. Syst. Man, Cybern.* **8** 460

[19] Mao J and Jain A K 1992 *Pattern Recogn.* **25** 173

[20] Haralick R M, Shanmugam K and Dinstein I 1973 *IEEE Trans. Syst. Man, Cybern.* **3** 610

[21] Hermes T, Klauck C, Krey W J and Zhang J 1995 *Proc. SPIE* **2420** 394

[22] Sakamoto H, Suzuki H and Uemori A 1994 *Proc. SPIE* **2185** 25

[23] Rui Y, Huang T S and Mehrotra S 1998 *Relevance Feedback Techniques in Interactive Content-Based Image Retrieval* (*Proc. SPIE*) **3312** 25

[24] Tsai D M and Lin B T 2002 *J. Mater. Eng. Performance* **20** 420

[25] Xiao B, Li J and Gao X B 2009 *Acta Electron. Sin.* **37** 2205 (in Chinese)

[26] Torre L D, Vrscay E R, Ebrahimi M and Barnsley M F 2009 *SIAM J. Imaging Sci.* **2** 470

[27] Wu X W, Jackson D J and Chen H C 2005 *Comput. Electron. Eng.* **31** 402

[28] Furao S and Hasegawa O 2004 *Signal Process. Image* **19** 393

[29] Kovacs T 2008 *Image Vision Comput.* **26** 1129

[30] Pang C Y, Zhou Z W and Guo G C 2006 *Chin. Phys.* **15** 3039

[31] Pang C Y, Zhou Z W, Chen P X and Guo G C 2006 *Chin. Phys.* **15** 618

[32] Long G L and Xiao L 2004 *Phys. Rev.* A **69** 052303

[33] Ren H P, Ping Z L, Bo W R G, Sheng Y L, Chen S Z and Wu W K 2003 *Chin. Phys.* **12** 610

[34] Atef M, William P and Mohamed S B 2010 *J. Electron. Imaging* **19** 1

[35] Duan L L, Liao X F and Xiang T 2010 *Commun. Nonlinear Sci. Numer. Simul.* Accepted

[36] Xu Y, Dong J T and Wang S H 2010 *Acta Phys. Sin.* **59** 7535 (in Chinese)