

Enhancing fractal image compression speed using local features for reducing search space

Keyvan Jaferzadeh¹ · Inkyu Moon¹ · Samaneh Gholami¹

Received: 3 January 2015 / Accepted: 4 May 2016 / Published online: 14 May 2016
© Springer-Verlag London 2016

Abstract The encoding step in full-search fractal image compression is time intensive because a sequential search through a massive domain pool has to be executed to find the best-matched domain for every range block. To afford a fair encoding time, immaterial domain–range block comparisons should be prevented. In this paper, a new local binary feature resemble to local binary patterns method is introduced. This single local feature is robust to noise and can exploit the general structure of the block. Concerning similarity between range–domain blocks, a criterion is allocated dynamically by measuring the pixel diversity among the range block pixels. To avoid redundant calculations, the distance of the general pattern is assessed by the Hamming distance utilizing a pre-computed table. Experimental results show that the presented approach can make FIC a lot faster as opposed to the full-search method and outperform some other identical methods while preserving the quality of the decoded images. Indeed, the proposed method can be utilized inside identical applications that want a specific block size or blocks comparing.

Keywords Fractal image compression · Local features · Hamming distance · Local binary pattern · Adaptive thresholding

1 Introduction

The reduction of data storage requirements is the main outcome of data compression methods. Furthermore, compression offers an attractive approach to decrease the communication cost while transferring substantial volumes of data through links via a relatively high effective utilization of the bandwidth of the available links. Due to the reduction in the data rate, the cost of communication can significantly make use of these methods.

The fractal image compression (FIC) algorithm has received considerable attention not only as a powerful image compression technique [1] but also for applications to different areas such as image restoration, medical image classification, watermarking, shape recognition, and face recognition [2–6]. Furthermore, the decoding of fractal encoded images is straightforward, fast, and very easy to implement. Besides, the resolution-free decoding property is another advantage of FIC, which means that the decoder can retrieve compressed images in different zooming scales. Microsoft Multimedia Encyclopedia (Microsoft Encarta), for example, used FIC to compress a large number of images, which practically shows that the properties of FIC make it suitable for multimedia applications [1].

The original idea of FIC was proposed by Barnsley [7] in which a considerable amount of redundancy existing in the images could be explored. To eliminate redundancies, Jacquin [8] proposed a partitioned iterated function system (PIFS) that contains contractive transformations for each image that together have a fixed point similar to that of the original image. This implies that by applying the transforms iteratively on an arbitrary initial image, the output image will converge to a fixed point similar to that of the original image. As such, storing these transformations means less space, and this can be considered an effective compression method [9].

✉ Keyvan Jaferzadeh
kjaferzadeh@chosun.kr

¹ Department of Computer Engineering, Chosun University,
309 Pilmun-daero, Dong-gu, Gwangju 501-759, South Korea

1.1 Brief review of previous efforts

Since the largest computation cost lies in the sequential search through the domain blocks, efforts toward reducing the range–domain block matching, eliminating unfitted domain blocks, and classifying domain blocks into similar subsets have centered on making the search faster [10, 11]. Since standard deviation (STD) is one of the important mathematic tools used in measuring the dispersion degree of a data set, previous efforts using STD showed advantages in accelerating encoding time in FIC [12–14]. Hybrid techniques like genetic algorithm and spatial correlation method [15], DCT coefficients and applying fuzzy clustering [16] can enhance encoding time significantly. To facilitate a high-speed domain pool search, binary matching technique [17], entropy-based encoding [18], and diagonal domain scaling while domain construction [19] are proposed. In addition to the previous efforts for the restriction of the search space, some “no search methods” have also been proposed [20–22], which are very fast for real-time applications. While “no search methods” are very fast, the reconstruction image quality is not sufficiently good, particularly for small images.

In this paper, a new local binary feature extractor much like local binary patterns is introduced. Considering this local feature, the proposed method can calculate dissimilarity between range and domain blocks by utilizing the simple, quickly, and efficient Hamming distance method as opposed to the costly least-square method. Similarity criterion is adaptively issued in line with the pixel dispersion degree calculated by STD value of each corresponding range block. Experimental results show that the adaptive criterion strategy not just excludes unnecessary domain–range block matching but preserves the quality of the decoded images. Additionally, to avoid redundant calculations, Hamming distances are stored in a pre-computed table with two entries.

Our key contributions are summarized as follows.

- We introduce a new operator resemble to local binary patterns (LBP) method to exploit local features. This operator can minimize noise by constructing a super-pixel using averaging technique.
- We introduce a new scanning method to enhance pattern discrimination by increasing the number of pixels. It turns out that the following scanning method can improve quality of the decoded images.
- Instead of measuring similarity between two features, we are interested in measuring degree of dissimilarity between two patterns by applying simple and fast Hamming distance technique.
- We introduce a dynamic weight assigning while pattern matching according to the pixel diversity within each range block.

- A pre-computed table can facilitate and accelerate measuring Hamming distances between two corresponding patterns of range and domain block.

The rest of this paper is organized as follows: In Sect. 2, a review of the basic FIC algorithm is presented. Section 3 describes the proposed approach and the similarities and differences between the proposed method and the standard local binary patterns. The scheme of proposed algorithm is described in Sect. 4. Experimental results and discussions are presented in Sect. 5, and finally, we conclude this paper with a brief summary in Sect. 6.

2 Review of full-search FIC algorithm

First, the original image f with size $N \times N$ should be divided into non-overlapping range blocks and overlapping domain blocks, which comprise the domain pool Q ; the size of the former is $C \times C$ and that of the latter is $B \times B$. In general, domain blocks could be of any size larger than the size of a range block, but in this study, it is double the size of the range block. Each such domain block must be spatially contracted (or decimated) before it is matched to one of the range blocks. All the range and domain blocks can be obtained by a sliding window method. The sliding window method is that two windows of $C \times C$ and $B \times B$ slide through the image f along the horizontal and vertical axes to partition the image f into range and domain blocks. Figure 1 shows the arrangement of range and domain blocks, and step length of sliding windows. Finally, for each range block $R_i (1 \leq i \leq N \times N / (C \times C))$, the algorithm searches Q to get the best similar domain block D_j and a contractive affine transformation T_i , which must assure that $T_i(D_j)$ has the maximum similarity (minimum distance) to R_i . Finally, we will obtain a set of contractive affine transformations $T = \{T_i, 1 \leq i \leq N \times N / (C \times C)\}$. For a set of transformations, considering the collage theorem [7], the union of transformations is contractive and

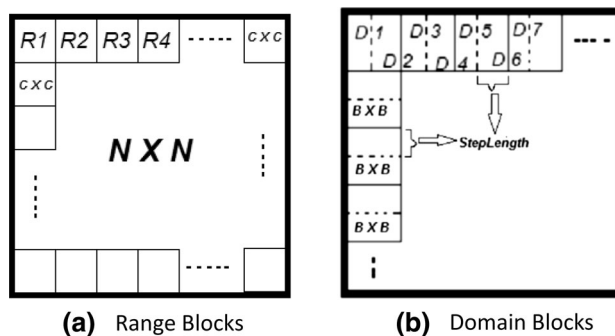


Fig. 1 Arrangement of range and domain blocks

the distance between their fixed point and the original image is arbitrarily small if each transformation is contractive. The position of domain block D_j and the parameters of transformation T_i constitute the fractal code of the range block R_i , and the fractal codes of all range blocks constitute the PIFS of the original image f . The decoding procedure is a simple iterative process, starting with any arbitrary initial image and iteratively applying inverse contractive affine transforms $T = \{T_i, 1 \leq i \leq N \times N / (C \times C)\}$ to the best-matched domain blocks. Because of the nature of fractal images, the union of inverse transformations can produce the original image from the compressed image [7–9]. The most time-consuming step of the encoding process is the searching for a domain block among Q , because every range block must be compared to all domain blocks to find the best one. But these range–domain block comparisons require a computationally expensive least-square measure (Eq. 1), which makes FIC infeasible for real-time applications.

$$Error(R_i, D_j) = \|s_i D_j + o_i I - R_i\|^2, \tag{1}$$

where s_i and o_i denote the optimal affine parameters obtained by the following:

$$s_i = \frac{\langle R_i - \bar{r}I, D_j - \bar{d}I \rangle}{\|D_j - \bar{d}I\|^2}, \tag{2}$$

$$o_i = \bar{r} - s_i \bar{d}, \tag{3}$$

where \bar{r} and \bar{d} denote the mean of range block R_i and that of domain block D_j , respectively.

3 Feature extraction

In image processing, the main goal of feature extraction is to obtain the most relevant information from the original image and represent it in a lower dimension. To say that, there are times that input data should be transformed into a reduced representation set of feature(s) since the input data to an algorithm are too large for further processing. This transformation from the input image into the set of features is so-called feature extraction. It is expected that the set of features can contain the relevant information from the input image if the extracted features are carefully chosen. There are previous efforts for dimension reduction in application of image processing and image understanding. Zecaho et al. [23, 24] proposed a new unsupervised feature selection technique to reduce the feature dimension. They try to identify the best subset of the most useful features by integrating cluster analysis along with sparse structural analysis. In case of FIC technique, we would like to extract some features that can avoid expensive least-square metric. Since in FIC we are interested in block matching, local features from each block

are desirable. This local feature should provide relevant information about the general structure of the block. Accordingly, dissimilar blocks can be ignored easily without calculating least-square metric. However, for the case of similar features, further processing can be done by the least-square metric. We believe that a good feature can distinguish one block from other non-similar blocks. This feature must be as robust as possible in order to prevent generating different feature codes for the similar blocks regarding the noise within block. Accordingly, we are interested only in local features for each single block. We also believe that the feature extraction step should be simple and quick since the main drawback in FIC is the expensive compression step and feature extraction step should not be a big overload to the whole compression technique.

3.1 Original local binary patterns

The local binary pattern texture operator was first introduced as a complementary measure intended for local image contrast by Ojala et al. [25]. The most important properties of the LBP operator in real-world applications are their invariance in opposition to monotonic gray-level changes and its particular computational simpleness. An LBP code is produced by multiplying the thresholded values with weights given by the corresponding pixels and summing up the end result. As the neighborhood consists of 8 pixels, throughout its earliest version, an overall total of $2^8 = 256$ distinct labels can be obtained according to the relative gray values of the center and the neighborhood pixels [26]. Figure 2 shows an original image of Lenna and the output of the LBP operator.

Although LBP have certain advantages as previously mentioned, it also holds two significant flaws, i.e., sensitivity to noise and tendency to characterize different structural textures with the same binary code, which causes a discriminability reduction. In order to overcome the mentioned weaknesses and make LBP more robust and stable, different strategies are proposed [27–29]. With these kinds of strategies, essentially, the local differences

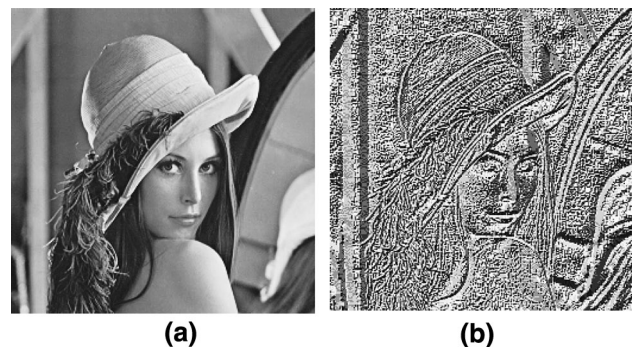


Fig. 2 a Original Lenna and b LBP output

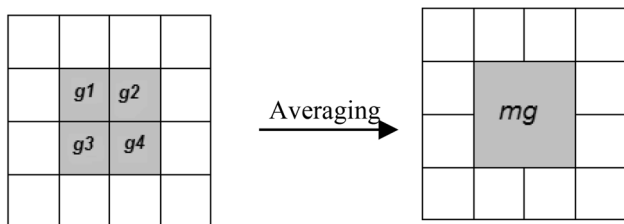


Fig. 3 New super-pixel instead of conventional central pixel

in addition to conventional gray value of the center pixel are enhanced as a way to increase discrimination in between different patterns.

3.2 Proposed local feature extraction operator

In this paper, a new local feature extractor similar to the standard LBP due to the block size property of FIC is proposed. The proposed strategy considers two essential changes that could manage the less noisy central pixel and adds more local features than the original LBP. The first step forms a super-pixel named mg by averaging four central pixels with two benefits: delivering the central pixel for the threshold value and improving the classification by averaging noisy pixels. In this paper, the range and the domain size are 4×4 and 8×8 , respectively. Accordingly, the suggested strategy to acquire the super-pixel is represented in Fig. 3. $g_1, g_2, g_3,$ and g_4 denote four central pixels; therefore, mg is:

$$mg = (g_1 + g_2 + g_3 + g_4)/4. \tag{4}$$

As discussed earlier, mg can provide a new super-pixel rather than the conventional central pixel. It can also diminish the noise effects simply by averaging the pixels. Second, rectangular neighborhoods are utilized rather than standard round ones. This rectangular strategy not only adds more pixels to our local feature but also avoids the circular position of conventional LBP [26]. In this study, the direction of picking up pixels is clockwise, orienting in the particular top-left pixel as shown in Fig. 4, but any arbitrary direction can be preferred. It is worth mentioning that different combinations of pixels for constructing super-pixel are used. Our findings show that the current combination can provide the best super-pixel regarding noise lowering. Likewise, 12-bit pattern can include adequate patterns regarding pattern discriminations aims comparing with the 8-bit conventional LBP (data not shown). After the above adjustments, a sequence of 12 bits, starting from p_0 to p_{11} , will be acquired. Eventually, the LF value for each block is acquired using the following:

$$LF = \sum_{u=0}^{11} Z(mg - p_u)2^u, \tag{5}$$

where $Z(u)$ denotes the thresholding function defined as follows:

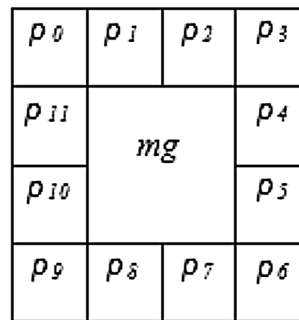


Fig. 4 Rectangular neighborhood instead of the circular one of standard LBP

$$Z(u) = \begin{cases} 1 & u \geq 0 \\ 0 & u < 0 \end{cases} \tag{6}$$

The suggested strategy has 2^{12} bits and thus explores 4096 textures.

4 Proposed FIC algorithm

The binary similarity and dissimilarity distances play a critical role in pattern analysis. Considering that the overall performance will depend on the selection of the appropriate measure, numerous binary similarity measures and distance measures have been proposed in various fields [30]. Generally, distance measures consider positive matches, negative matches, and mismatches for the final judgment. In this work, regarding measuring the similarity between the corresponding range and domain blocks, the Hamming distance is preferred since it only takes number of mismatches. Our experimental results show that among the well-known binary similarity measures (Jaccard, Dice, Euclid, simple matching coefficient, and Czekanowski Index), Hamming distance outperforms in both running time and the decoded image quality (data not shown). Suppose that V_1 and V_2 are representing the two binary feature vectors and n the number of attributes or length of the feature vectors. Definitions of Hamming distance can be expressed by the following:

$$HMD(V_1, V_2) = \sum_{i=1}^n Xor(V_{1,i}, V_{2,i}) \tag{7}$$

We calculated Hamming distances for the best-matched range–domain blocks using full-search method (Table 1). The image size is 256×256 , the range size is 4×4 , and the domain size is 8×8 ; therefore, the total number of range blocks is 4096, and the domain pool has 64,001 blocks.

Table 1 shows that 4041 (99 %) out of 4096 of the matched blocks for the Lenna image have $HMD \leq 5$.

Table 1 Matched range–domain blocks and their HMD values for *StepLength* = 1

	HMD = 0	HMD = 1	HMD = 2	HMD = 3	HMD = 4	HMD = 5	HMD = 6	HMD = 7	HMD = 8
Lenna	848	1150	992	637	289	125	42	11	2
Pepper	756	1135	1046	678	308	129	33	9	5
Crowd	837	1177	921	599	307	140	67	26	15
Cameraman	692	996	1016	745	366	193	75	20	8
Goldhill	728	1168	1007	683	328	130	35	14	3

Table 2 PSNR and encoding time for full search and the discussed strategy

	Lenna	Pepper	Crowd	Cameraman	F16	Goldhill
<i>PSNR value</i>						
Full search	31.83	33.3	30.37	31.02	31.32	32.03
HMD ≤ 5	31.83	33.27	30.36	31.0	31.3	32.02
<i>Encoding time (s)</i>						
Full search	58.81	61.01	60.25	63.01	64.13	62.34
HMD ≤ 5	30.86	31.59	31.15	30.21	29.94	31.06

Considering Table 1, we infer that more than 98 % of the matched blocks have HMD between 0 and 5. This indicates that domain blocks having $HMD \leq 5$ corresponding to each range block shall be considered in the costly least-square measure, and the rest of the domains can be excluded. To evaluate the end results connected to the excluding irrelevant range–domain block matching, we adjusted original FIC. While the domain pool search, if the HMD between the range block and domain block of interest exceeds five, the domain block is disregarded and the search proceeds with the next domain block. Otherwise, least-square (Eq. 1) is performed to determine the range–domain block match. Excluding these domain blocks does not degrade image quality considerably (Table 2).

As stated previously, eliminating irrelevant range–domain blocks matching speeds up the search albeit at the expense of small loss of quality, and the main concern is maintaining a good balance for different images. If the HMD threshold is large, a lot more domain blocks will be searched using the expensive least-square (Eq. 1), leading to a longer encoding time. However, better reconstructed image quality will be retrieved (Fig. 5). On the other hand, reducing the HMD will probably speed up the encoding process, nonetheless, at the expense of lower quality because some of the qualified domain blocks will no longer be available. Considering the above discussion, a proper threshold for HMD (only $HMD \leq 5$) to balance the encoding time and keep fidelity is critical.

To cope with the adaptive threshold issue, we discovered visual properties and patterns in sample images. We think that a smaller HMD value should be assigned to smooth blocks and a larger value should be assigned to

textured range blocks. Since in the proposed strategy we need to deal with different kind of range blocks (smooth to textured one), an intelligent HMD assignment strategy based on the STD value of each range is proposed. STD is one of the best means of measuring diversity and contrast (root-mean-square contrast) in image data. It is a statistical solution to define certain properties of blocks, such as pixel dispersion and contrast value. While a smaller value of STD shows a smooth and low-contrast region, a higher value indicates unsmoothed, high-contrast, and textured blocks. The definition of STD is as follows:

$$STD = \sqrt{\frac{\sum_{i=1}^C \sum_{j=1}^C (P_{ij} - \bar{r})^2}{C \times C}}, \tag{8}$$

where $P_{i,j}$ denotes the pixel value at position (i, j) and \bar{r} represents the mean value of the range block. If R is a range block and its STD is small, it means that the pixel values of R fluctuate little. Since R is a smooth block, we can search domains with smaller HMD. On the other hand, when STD is high, we think that the best-matched domain block of R has a larger HMD. So for each range block R_i and the domain blocks in Q , we think that the best-matched domain block should fall into the following set:

$$\{D(R_i) \in Q | HMD(D, R_i) \leq THMD\}, \tag{9}$$

where HMD is Hamming distance and $THMD$ is the control threshold. Besides utilizing STD for assigning an adaptive value for HMD, a pre-computed lookup table for avoiding redundant HMD calculations, namely *HMDTable*, is utilized. This lookup table replaces runtime computation with a simpler indexing operation. The *HMDTable* has two LF

Fig. 5 PSNR value versus Hamming distance

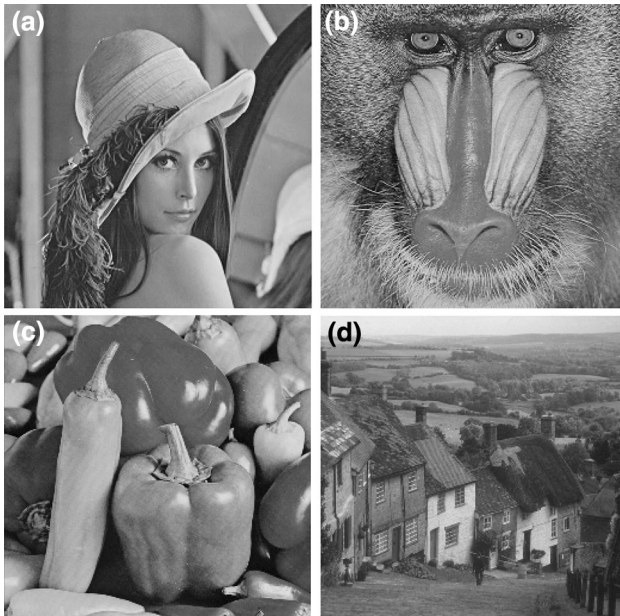
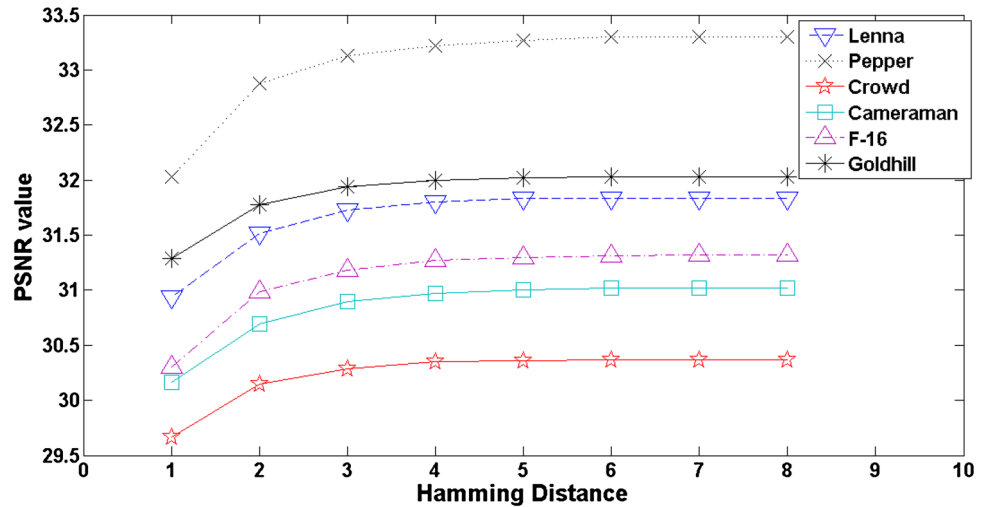


Fig. 6 Original images **a** Lenna, **b** Baboon, **c** Pepper, and **d** Goldhill

value entries of range and domain blocks, which return the HMD for the entries.

The proposed encoding algorithm consists of the following steps:

1. Partition the image into non-overlapped range blocks of R_i . For each block, calculate its LF value using Eq. 5 and STD value using Eq. 8; the former is called LFR_i , and the latter is called $STDR_i$.
2. Partitioning an image into overlapping domain blocks of D_j , calculate their LF values and name them LFD_j .
3. Find the maximum and minimum values of all $STDR_i$ s, and label them $MaxSTD$ and $MinSTD$, respectively.

4. Consider the criterion for $x = 0$ to 5

$$Cr_x = \begin{cases} Cr_{x-1} + (MaxSTD - MinSTD)/5 & x > 0 \\ MinSTD & x = 0 \end{cases} \quad (10)$$

5. While there is uncovered R_i , do the following:
 - a. If $(Cr_0 \leq STDR_i \leq Cr_1)$ $THMD = 1$;
 - b. Else if $(Cr_1 < STDR_i \leq Cr_2)$ $THMD = 2$;
 - c. Else if $(Cr_2 < STDR_i \leq Cr_3)$ $THMD = 3$;
 - d. Else if $(Cr_3 < STDR_i \leq Cr_4)$ $THMD = 4$;
 - e. Else $THMD = 5$;

6. While performing range–domain block matching using Eq. 1, consider only those domains that can satisfy Eq. 9.

The proposed fractal image encoding algorithm is divided into three parts: the block-construction phase (Steps 1 and 2), marginal values calculation (Steps 3 and 4), and the encoding phase. In the block construction phase, the original image f is partitioned into non-overlapping range blocks $C \times C$ and overlapping $B \times B$ domain blocks. The step length of domain blocks is denoted by $StepLength$. Then, the STD values of range blocks are calculated, and maximum and minimum values are picked up. According to these STD numbers, five marginal values (Cr_0 – Cr_4) for separating Hamming distances are calculated. With the encoding phase, according to the STD value of every range block R_i , an effective $THMD$ will be assigned (only the domain blocks in Q whose HMD value can fulfill Eq. 9 are good candidates). Then the search for the best-matched domain block of R_i is carried out in the domain block set $D(R_i)$ (see Eq. 9).

The primary disadvantage of fractal encoding is the large amount of time required to search for the best range–

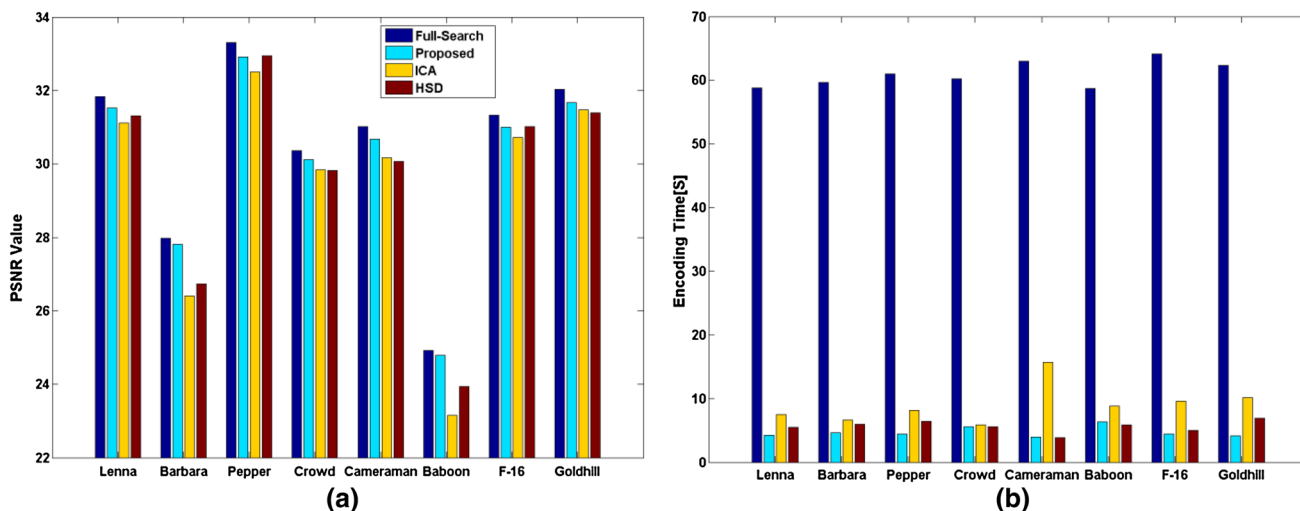


Fig. 7 Comparing different methods $StepLength = 1$; **a** PSNR values and **b** encoding time

domain block match from the domain pool. Through the domain pool search, if the HMD between the range block and domain block of interest exceeds a specific threshold, the domain block will be ignored and the search proceeds with the next domain block. Otherwise, Eq. 1 matching operation is carried out to look for the best range–domain block match. As stated previously, selecting adaptive HMD values can lead to an acceptable speed and relatively high reconstructed image quality. That is, range blocks with higher STD (high-contrast blocks) demand comparing against more domain blocks for discovering the best-matched block. Therefore, selecting a higher HMD value will include more domain blocks in the block matching step. On the other hand, low-contrast range blocks (smoothed blocks) with lower STD can be easily matched with a small number of domain blocks. Thus, selecting a smaller value of HMD will exclude a large number of immaterial domain blocks. This threshold, of course, is an adaptively changing value.

5 Experimental results and discussion

Several experiments were conducted to demonstrate the performance of the proposed method. The proposed algorithm was compared with different methods, including the full-search method, intelligent classification algorithm (ICA) based on STD [13], and hybrid method using STD and DCT values (HSD) [14]. Different graphs and tables are illustrated that include the PSNR value and the encoding time. The eight images used for this experiment are Lenna, Babbon, Peppers, Goldhill (Shown in Fig. 6), Crowd, Cameraman, F16 and Barbara. All the experiments were executed on a PC with Intel Core i5, 2.5-GHz

processor, 3-GB RAM, and 32-bit Windows Seven operating system. All methods are implemented in MATLAB. In this study, the scale factor of S_k was set to a fixed value of unity for all the considered methods. It is shown that fixed scale factor value can accelerate the encoding step manifold while preserving the quality of the reconstructed image [12]. The STD limit for searching domain blocks for the method presented in [13] was set as 0.2, and K for the hybrid method is 300 and T_s is 3.0 [14]. As mentioned earlier, the image, range, and domain size are 256×256 , 4×4 , and 8×8 , respectively; therefore, the total number of range blocks was 4096, and the domain pool had 64,001 blocks. The compression ratio is 1.5 bpp because 24 bits are needed to store the matched domain block position, 8 bits for storing \bar{r} , and $S_k = 1$. Indeed, fractal parameters are stored as raw data in the output file without any other data compression technique. The quality measurement of the retrieved image is the peak signal-to-noise ratio (PSNR) defined as:

$$PSNR = 20 \times \log_{10} \left(\frac{MAX_f}{\sqrt{MSE}} \right) \tag{11}$$

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \|f(i,j) - f'(i,j)\|^2 \tag{12}$$

where $MAX_f = 255$, f is the original picture and f' is the decoded image.

A comparison of our approach’s PSNR value with the other method is graphically illustrated in Figs. 7a and 8a. In case of $StepLength = 1$ (Fig. 7a), we can see that the proposed method can achieve a reasonable image quality and the PSNR decay compared to the full-search method can be disregarded. The proposed method outperforms other methods in all sample images, particularly in the case of Baboon and Barbara due to the many unsmoothed

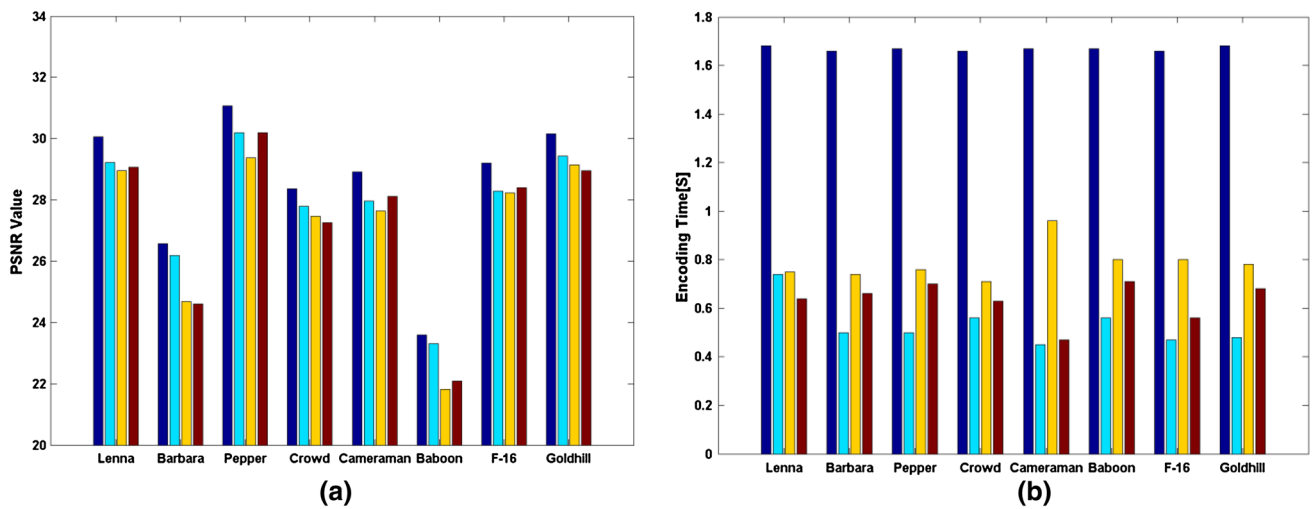


Fig. 8 Comparing different methods $StepLength = 4$; **a** PSNR values and **b** encoding time

Table 3 Sum of range blocks within each threshold value

	$THMD = 1$	$THMD = 2$	$THMD = 3$	$THMD = 4$	$THMD = 5$
Lenna	3191	625	214	50	16
Barbara	2266	1075	587	156	12
Pepper	3244	464	256	104	28
Crowd	2457	1019	479	115	26
Cameraman	3358	315	216	172	35

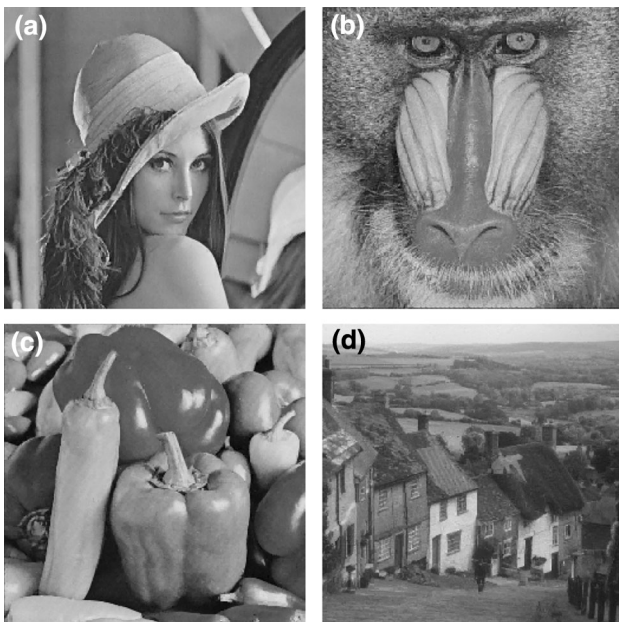


Fig. 9 Decoded sample images using the proposed method: **a** PSNR = 31.52; encoding time = 4.23 s, **b** PSNR = 24.79; encoding time = 6.37 s, **c** PSNR = 32.92; encoding time = 4.43 s, **d** PSNR = 31.67; encoding time = 4.18 s

blocks (Fig. 7a). In terms of encoding time (Fig. 7b), the proposed method is faster than other similar methods. In the case of $StepLength = 4$ in most of the cases, our

method has better quality comparing to the similar methods (Fig. 8a). Considering encoding time for $StepLength = 4$, the proposed method is still better than other methods, except Lenna (Fig. 8b).

Table 3 shows that the sum of the range blocks for each $THMD$ differed in value, as stated in the proposed encoding algorithm. From Table 3, we infer that for images that have more unsmoothed blocks, $THMD = 2$ increases and $THMD = 1$ decreases. For example, in the case of Barbara or Crowd that has many unsmoothed patterns (high-contrast areas), there is a higher number for $THMD = 2$, while in the case of Cameraman, this value is small.

We also show the decoded images of Lenna, Baboon, Pepper, and Goldhill that were encoded using the proposed strategy (Fig. 9a–d). Original images are shown in Fig. 6.

These images show that the proposed approach can preserve image quality because the PSNR decay compared to the full-search method is very small. Indeed, they show a considerable acceleration of the encoding step.

6 Conclusions

Fractal image compression and its related applications have been restricted because of the intensive encoding time. Eliminating unqualified range-domain block matching

usually speeds up the search albeit at the expense of some quality loss. The novelties of the proposed approach, that are extracting the local features, Hamming distance for measuring the variations between two different bit streams, and intelligent threshold assignment using STD, can significantly exclude a large number of unnecessary block matching operations. A table with two entries is defined to avoid redundant calculation of similarities. Another important advantage of this method is that it does not have considerable overload calculation because local features and STD values are obtained simply during range and domain block formation. The proposed method decreases the encoding time significantly as compared to the full-search method, at a PSNR decay of <0.30 . Because of the above-mentioned strategies, the proposed method can achieve a good speed-up as compared to similar benchmarks.

Acknowledgments This research was supported by Basic Science Research Program through the National 356 Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning 357 (NRF-2015R1A2A1A10052566).

References

- Barnsley M, Saupe D, Vrscay E (2002) *Fractals in multimedia*. Springer, New York
- Bocchi L, Coppini G, Nori J, Valli G (2004) Detection of single and clustered micro-calcifications in mammograms using fractals models and neural networks. *Med Eng Phys* 26(4):303–312
- Rongrong N, Qiuqi R, Cheng H (2005) Secure semi-blind watermarking based on iteration mapping and image features. *Pattern Recogn* 38(3):357–368
- Daraee F, Mozaffari S (2014) Watermarking in binary document images using fractal codes. *Pattern Recogn Lett* 35:120–129
- Gdawiec K, Domanska D (2011) Partitioned iterated function systems with division and a fractal dependence graph in recognition of 2D shapes. *Int J Appl Math Comput Sci* 21(4):757–767
- Kouzani AZ (2008) Classification of face images using local iterated function systems. *Mach Vis Appl* 19(4):223–248
- Barnsley M (1988) *Fractal everywhere*. Academic, New York
- Jacquin AE (1992) Image coding based on a fractal theory of iterated contractive image transform. *IEEE Trans Image Process* 81:18–30
- Fisher Y (1995) *Fractal image compression: theory and applications*. Springer, New York
- Kovacs T (2008) A fast classification based method for fractal image encoding. *Image Vis Comput* 26:1129–1136
- Duh DJ, Jeng JH, Chen SY (2005) DCT based simple classification scheme for fractal image compression. *Image Vis Comput* 23:1115–1121
- Tong CS, Pi M (2010) Fast fractal image encoding based on adaptive search. *IEEE Trans Image Process* 10:1269–1277
- Xianwei W, Jackson DJ, Chen HC (2005) A fast fractal image encoding method based on intelligent search of standard deviation. *Comput Electr Eng* 31:402–421. doi:10.1016/j.compeleceng.2005.02.003
- Wang X, Zhang D, Guo X (2013) Novel hybrid fractal image encoding algorithm using standard deviation and DCT coefficients. *Nonlinear Dyn* 73(1–2):347–355. doi:10.1007/s11071-013-0790-2
- Wang XY, Li FP, Wang SG (2009) Fractal image compression based on spatial correlation and hybrid genetic algorithm. *J Vis Commun Image Represent* 20:505–510. doi:10.1016/j.jvcir.2009.07.002
- Jaferzadeh K, Kiani K, Mozaffari S (2012) Acceleration of fractal image compression using fuzzy clustering and discrete-cosine-transform-based metric. *Image Process IET* 6(7):1024–1030. doi:10.1049/iet-ipr.2011.0181
- Samavi S, Habibi M, Shirani S, Rowshanbin N (2010) Real time fractal image coder based on characteristic vector matching. *Image Vis Comput* 28(11):1557–1568. doi:10.1016/j.imavis.2010.03.011
- Salarian M, Nadernejad E, Naimi H (2013) A new modified fast fractal image compression algorithm. *Imaging Sci J* 61(2):219–231
- Tang G, Shuang W, Yan Z (2012) An improved fast fractal image coding algorithm. In: *Proceedings of 2nd international conference on computer science and network technology, ICCSNT*, pp 730–732
- Furaoa S, Hasegawa O (2004) A fast no search fractal image coding method. *Signal Process Image Commun* 19:393–404
- Wang XY, Wang SG (2008) An improved no-search fractal image coding method based on a modified gray-level transform. *Comput Graph* 32:445–450
- Wang XY, Wang YX, Yun JJ (2010) An improved no-search fractal image coding method based on a fitting plane. *Image Vis Comput* 28:1303–1308
- Zecaho L et al (2015) Robust structured subspace learning for data representation. *IEEE Trans Knowl Data Eng* 37(10):2085–2098
- Zecaho L et al (2014) Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Trans Pattern Anal Mach Intell* 26(9):2138–2150
- Ojala T, Pietikäinen M, Harwood D (1994) Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. *Int Conf Pattern Recogn* 1:582–585
- Pietikäinen M, Hadid A, Zhao G, Ahonen T (2011) *Computer vision using local binary patterns*. Springer, London
- Guo Z, Zhang L, Zhang D (2010) A completed modeling of local binary pattern operator for texture classification. *IEEE Trans Image Process* 19(6):1657–1663
- Zhao Y, Hu RX, Min H (2013) Completed robust local binary pattern for texture classification. *Neurocomputing* 106:68–76
- Backes A (2013) A new approach to estimate lacunarity of texture images. *Pattern Recogn Lett* 34:1455–1461
- Choi SS, Cha SH (2010) A survey of binary similarity and distance measures. *J Syst Cybern Inf* 8:43–48



Keyvan Jaferzadeh is a Ph.D. student at the Computer Engineering Department of Chosun University. He received his B.S. in Software Engineering and M.S. in Mechatronic Engineering in 2006 and 2010 in Iran, respectively. He has authored several other papers, and his current research interests include image processing, digital holography, and image compression.



Inkyu Moon received his B.S. in Electronics Engineering from SungKyunKwan University, Korea, in 1996, and his Ph.D. in Electrical and Computer Engineering from University of Connecticut in 2007. He joined Chosun University, Korea, in 2009, and is currently an Associate Professor at the School of Computer Engineering. His research interests include digital holography, biomedical imaging, and optical information processing. Dr. Moon is a

member of IEEE, OSA, and SPIE.