J. Vis. Commun. Image R. 25 (2014) 1835–1841

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Genetic algorithm based on discrete wavelet transformation for fractal image compression

Ming-Sheng Wu *

Department of Electrical Engineering, Cheng Shiu University, Kaohsiung, Taiwan

ARTICLE INFO

Article history:
Received 5 May 2014
Accepted 28 August 2014
Available online 16 September 2014

Keywords:
Fractal image compression
Partitioned iterated function system
Discrete wavelet transformation
Dihedral transformation
FIC using DWT
Evolutionary algorithm
Genetic algorithm
GA based on DWT

ABSTRACT

In this paper, a genetic algorithm (GA) based on discrete wavelet transformation (DWT) is proposed to overcome the drawback of the time-consuming for the fractal encoder. First, for each range block, two wavelet coefficients are used to find the fittest Dihedral block of the domain block. The similar match is done only with the fittest block to save seven eighths redundant MSE computations. Second, embedding the DWT into the GA, a GA based on DWT is built to fast evolutionary speed further and maintain good retrieved quality. Experiments show that, under the same number of MSE computations, the PSNR of the proposed GA method is reduced 0.29 to 0.47 dB in comparison with the SGA method. Moreover, at the encoding time, the proposed GA method is 100 times faster than the full search method, while the penalty of retrieved image quality is relatively acceptable.

1. Introduction

Fractal image compression (FIC) is a time-consuming encode algorithm. The original idea was proposed by Barnsley and Demko in 1985 [1] and the practical FIC algorithm was not implemented until 1992 by Jacquin [2]. The reason of the FIC spends overlong time on encode is that, based on the self-similarity property of real life images and Partitioned Iterated Function System (PIFS) [3,4], each range block must searching for the best matched block from the large domain pool. A large number of the redundant similar computations will slow down the encoding speed of the FIC. Therefore, the main research direction for FIC is focused on how to reduce the encoding time under the premise of maintaining retrieve image quality.

In the past, many encoding algorithms were presented to speedup the fractal encoder. In 2004, Truong et al. [5] proposed a fast fractal image compression using spatial correlation. The method limits the search space for the current range block on the neighborhoods of the matched domain blocks of the neighboring range blocks by utilizing the spatial correlations between neighboring blocks in both the domain pool and the range pool. Compared to full search method, the method achieves a 2.6 times speedup ratio. In the same year, Furao's no search method [6] incorporates the adjusted quad-tree scheme into the no-search scheme to improve coding fidelity significantly. In 2005, Duh et al. [7] classifies the blocks into three classes by using DCT. The classifier limits the similar match can be done only when the range block and domain block belong to the same class. Hence, the method can achieve 3 times speedup ratio in comparison to full search method. Wang et al. [8] in 2009 combine quad-tree framework, neighbor search, and asymptotic strategy to implement a fast coding method. Then Wang et al. [9] in 2010 proposed further a no-search fractal image coding method based on a fitting plane to improve the Furao's no search method. Compared to Furao's no search method, the compression ratio, the quality and encoding time are all improved greatly. As discussed above, the Fractal coding techniques can be roughly categorized into classification [7,10–14], quad-tree [6,8,15,16], no search [6,9,16], and spatial correlation techniques [5,14,17,18], etc.

Recently, there are evolutionary algorithms [17,19–23] used to solve the problem of the encoding of the FIC, in which genetic algorithm (GA) is the one of the most interested methods to the researchers. Traditionally, there are two chromosome formations are used to carry out the evolution of the GA algorithm. The first chromosome formation is composed of the x-coordinate and y-coordinate of the domain block in an image and Dihedral index. Such GA method can reduce indeed the computational load substantially, because each chromosome represents a Dihedral transformation block of the domain block and only the transformed block is taking to do the similar match with the range block. But

* Address: Department of Electrical Engineering, Cheng Shiu University, No. 840 Chengcing Rd., Niaosong Dist., Kaohsiung 83347, Taiwan.
E-mail address: sheng@csu.edu.tw

http://dx.doi.org/10.1016/j.jvcir.2014.09.001
1047-3203/

the disadvantage is that the landscape of the search space is too complicated to find the good matched solution. Hence, the improvement of the retrieved quality is limited. The second chromosome formation is formed by x-coordinate and y-coordinate of the domain block in an image. The landscape of the search space of the GA method is smooth and easy to find the near optimal solution. Such chromosome formation indeed can achieve good retrieved quality. But the drawback is each chromosome stands for a domain block, each range block must do the similar match with all the eight Dihedral transformation blocks of the domain block. A large number of MSE computations will slow down the speed of the encoder. Therefore, the trade-off between the retrieved image quality and encoding speed exists in the two chromosome formations.

In this paper, we propose a GA based on DWT to overcome the trade-off problem. First, a FIC using DWT is addressed to eliminate redundant MSE computations. For each range block, the method uses two wavelet coefficients to find the fittest Dihedral block from the eight Dihedral blocks of the domain block. The range block does the similar match only with the fittest Dihedral block. The other seven Dihedral blocks are discarded. Thus seven eighths redundant MSE computations will be ignored to achieve the goals of accelerating the fractal encoder and maintaining the retrieved image quality. Second, embedding the DWT technique into the GA, a GA based on DWT is implemented. The evolution speed of the proposed GA method will be faster than that of the traditional GA method since the length of the chromosome is shorter. This is because the fittest Dihedral block of the domain block has been determined and thus not required in the chromosome. Finally, the proposed GA method also attempts to compare with the other genetic method to demonstrate the performance of the proposed GA method.

In the next section, the theoretical basis of fractal image compression is outlined. The FIC using DWT is given in Section 3. Embedding the DWT into the GA, the proposed GA method is introduced in Section 4. In Section 5, the experimental results of comparing the proposed GA method with the full search and the other GA methods are provided. A conclusion is stated in Section 6.

2. Fractal image encoding

The fundamental idea of the fractal image compression is coming from the PIFS. For a original gray level image f of size $m \times m$, let the range pool R be the set of $(m/n)^2$ non-overlapping blocks of size $n \times n$ in the image f. For obeying the Contractive Mapping Fixed-Point Theorem, the domain block must exceed 1 times than the range block in size. Let the contraction ratio of the fractal coding be 2. Hence, the domain pool D is the set of $(m-2n+1)^2$ overlapping blocks of size $2n \times 2n$ in the image f. Based on the local self-similarity property in a nature image, each range block v from the R searches for the most similar block in the domain pool D to construct the fractal affine transformation. The parameters representing the fractal affine transformation will form the fractal compression code of v.

To encode an image, the domain block of the size $2n \times 2n$ must be sub-sampled to $n \times n$ such that its size is the same as v and denotes as u. In addition to the position of the domain block, the fractal affine transformation includes the Dihedral index d of the domain block. Hence, all the eight Dihedral transformation blocks $u_k{:}k = 0, 1, \ldots, 7$ of the u must do the similarity match with v to find the optimal one. The eight Dihedral transformation blocks are generated by the eight transformations $T_k{:}k = 0, 1, \ldots, 7$, respectively, which expressed by the matrices in Table 1, in which the origin of u is assumed to locate at the center of the block. An illustration of the eight Dihedral transformations is depicted in Fig. 1 [20]. T_0 takes the origin block u. Hence $u_0 = u$. T_1 and T_2 are the flip of u with respect to horizontal and vertical lines, respectively. T_3 is the flip of u with respect to both horizontal and vertical lines. T_4, T_5, T_6, and T_7 are the flip of the u_0, u_1, u_2, and u_3 along the main diagonal line, respectively.

Table 1
The 8 transformations in the Dihedral group.

T_0	T_1	T_2	T_3
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
T_4	T_5	T_6	T_7
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$

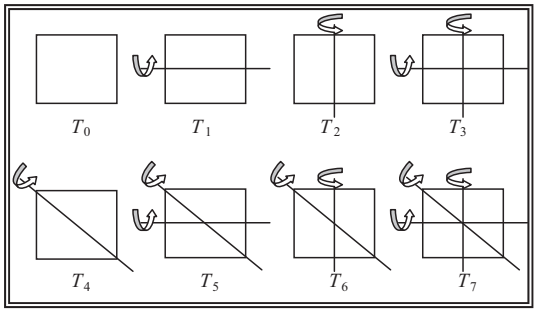

Fig. 1. The diagram of eight transformations in the Dihedral group.

The fractal affine transformation also includes the contrast scaling p and the brightness offset q of the transformed blocks. Thus the fractal affine transformation φ of $u(x,y)$ can be expressed as

$$\varphi\begin{bmatrix} x \\ y \\ u(x,y) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & p \end{bmatrix}\begin{bmatrix} x \\ y \\ u(x,y) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ q \end{bmatrix}, \tag{1}$$

where the sub-matrix $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ represents the one of eight Dihedral transformations in Table 1 and (t_x,t_y) is the coordinate of the domain block in the domain pool. For each domain block in the domain pool, the eight MSE computations between its eight Dihedral transformation blocks and v are done to find the index d of the best matched transformation block such that

$$d = \arg\min\{\|p_k \cdot u_k + q_k - v\| : \ k = 0, 1, 2, \cdots, 7\}.$$

Here, p_k and q_k can be computed directly by

$$p_k = \frac{[N\langle u_k, v\rangle - \langle u_k, \vec{1}\rangle\langle v, \vec{1}\rangle]}{[N\langle u_k, u_k\rangle - \langle u_k, \vec{1}\,\rangle^2]}, \quad q_k = \frac{1}{N}[\langle v, \vec{1}\rangle - p_k\langle u_k, \vec{1}\rangle],$$

where N is the number of pixels of the range block and $\vec{1} = [1 \quad 1 \quad \cdots \quad 1]^T$.

As u runs over all the $(m-2n+1)^2$ blocks in the domain pool, the best domain block is obtained. The parameters, including the coordinate (t_x,t_y), the Dihedral index d, contrast scaling p, and brightness offset q, constitute the fractal compression code of v. Finally, as v runs over all $(m/n)^2$ blocks in the range pool, the encoding process is completed. Such method is referred to as the full search method. There are $(m-2n+1)^2$ domain blocks and $(m/n)^2$ range blocks, and together with the eight Dihedral transformation, the amount of MSE computation is therefore $(m-2n+1)^2 \times 8 \times (m/n)^2$. For the case of 256×256 image with 8×8 coding unit, the amount of MSE computations is $(256-15)^2 \times 8 \times (1024)^2 = 475,799,552$.

To decode, the compression codes of $(m/n)^2$ range blocks make up the affine transformations. According to Contractive Mapping Fixed-Point Theorem and Collage Theorem [24], we select an arbitrary initial image and perform the affine transformations recursively. The process will not stop until some criterion is met.

3. The fractal encode algorithm using discrete wavelet transformation

In this section, a fractal image compression (FIC) using discrete wavelet transformation (DWT) is introduced. For each range block, two discrete wavelet coefficients are used to determine which Dihedral block among the eight Dihedral blocks of the domain block is fittest. The range block does the similar matching only with the fittest Dihedral block to achieve the number of one eighth MSE computations in comparison with the full search method.

Essentially, the DWT of an image is a two-dimension DWT. Fig. 2 shows the n-level DWT decomposition of an image block, in which the size of the image block is $2^n \times 2^n$ and *LLn*, *LHn*, *HLn*, and *HHn* are real number. *LLn* is called direct coefficient and is the average value of all the pixel of the image block. It reveals the low frequency information of the image block. *LHn*, *HLn*, and *HHn* are called detail coefficients, which depict the contour information of the horizontal edge, the vertical edge, and the diagonal edge of the image block, respectively. Especially, the magnitudes of *LHn* and *HLn* reflect respectively the intensity variation between the upper half and lower half and the intensity variation between the left half and right half for the image block. The signs of them show respectively the variation of the brightness from bright to dark at vertical and horizontal directions. If $LHn > 0$, the brightness at the upper is higher than the one at the lower. On the contrary, if $LHn < 0$, the brightness at the upper is smaller than the one at the lower. Similarly, If $HLn > 0$, the brightness at the left half is higher than the one at the right half. Opposite, if $HLn < 0$, the brightness at the left half is smaller than the one at the right half. Moreover, Table 2 shows the variation of magnitudes and signs of *LHn* and *HLn* for an image block through eight Dihedral transformations. At the second column, the *LHn* and *HLn* of the original image block are *A* and *B*, respectively. The fifth column shows the coefficients *LHn* and *HLn* of the transformed block obtained by T_2 transformation on the original image block, which are – *A* and *B*, respectively. Compared the *LHn* and *HLn* of the two image blocks, their absolute magnitudes are the same and only the sign of the *LHn* is opposite. The table indicates that when an image block is carried out the

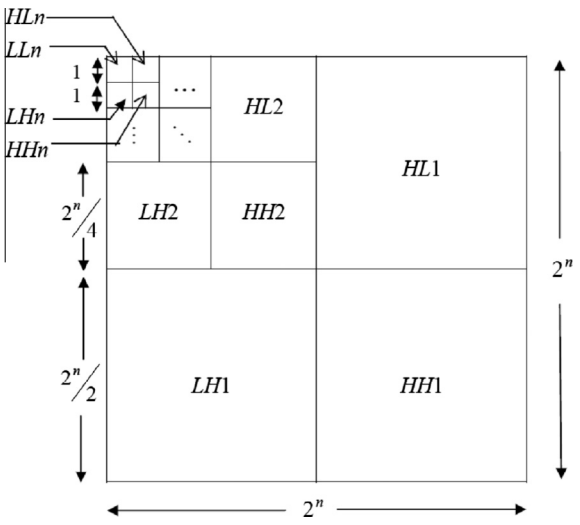

Fig. 2. The n-level decomposition of an image block of size $2^n \times 2^n$.

Table 2
The variation of magnitude and sign of *LHn* and *HLn* of an image block after eight Dihedral transformations.

	Origin image	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
LHn	*A*	*A*	*A*	$-A$	$-A$	*B*	$-B$	*B*	$-B$
HLn	*B*	*B*	$-B$	*B*	$-B$	*A*	*A*	$-A$	$-A$

Dihedral transformation, the *LHn* and *HLn* of the transformed block with respect to the original image block only changed their sign and/or exchanged their magnitude.

For the two image blocks f_1 and f_2, their *LHn* and *HLn* are LHn_1 and HLn_1 and LHn_2 and HLn_2, respectively. The two image blocks are the similar if the signs of their *LHn* and *HLn* are the same and the absolute magnitudes of their *LHn* and *HLn* are very near, i.e., $sign(LHn_1) = sign(LHn_2)$, $sign(HLn_1) = sign(HLn_2)$, $|LHn_1| \approx |LHn_2|$, and $|HLn_1| \approx |HLn_2|$. In other words, under such a condition, the two image blocks have the same edge direction and the same varied direction of the brightness. Therefore for the given range block r and domain block d, we must find a best Dihedral transformation $T_k, 0 \leqslant k \leqslant 7$ such that $T_k(d)$ is the best match of r. In this case, the *LHn* and *HLn* of both $T_k(d)$ and r conform the relationship stated above. The $T_k(d)$ is most similar to r and the T_k will be the optimal Dihedral transformation between them. Table 3 lists the optimal Dihedral transformation relationship between the range block r and domain block d by using *LHn* and *HLn*, in which *LHnr* and *HLnr* are the *LHn* and *HLn* of the r, respectively, and *LHnd* and *HLnd* are the *LHn* and *HLn* of the d, respectively. For example, if the *LHnr* and *HLnr* of the r have the relation shown on row 4 of Table 3, i.e., $LHnr < 0$, $HLnr > 0$, and $|LHnr| > |HLnr|$, the r has obviously horizontal edge and the brightness at the lower and the left half is higher than the brightness at the upper and the right half, respectively. A d is taken to match with the r and the relation of *LHnd* and *HLnd* is described by column 6 of Table 3, i.e., $LHnd > 0$, $HLnd > 0$, and $|LHnd| < |HLnd|$. The d has obviously vertical edge and the brightness at the upper and the left half is higher than the brightness at the lower and the right half, respectively. According to the indication of Table 3, the d is taken T_5 transformation. Table 2 shows that when the block d is transformed using T_5, the *LHnd* and *HLnd* of the transformed block $T_5(d)$ are equal to the $-HLnd$ and *LHnd* of d, respectively, in which the transformed block $T_5(d)$ has the condition of $LHnd < 0$, $HLnd > 0$, and $|LHnd| > |HLnd|$. The transformed block $T_5(d)$ will also have obviously horizontal edge and the brightness at the lower and the left half is higher than the brightness at the upper and the right half, respectively. Hence for all eight Dihedral transformation blocks of the d, the transformed block $T_5(d)$ is the most similar to the r at the visual viewpoint. The r does the similar match only with the optimal transformed block $T_5(d)$. The others seven transformed blocks $T_k(d)$, $k = 0, 1, 2, 3, 4, 6, 7$ will be discarded. Thus, compared to the full search method, the proposed method needs only one eighth MSE computations and the resulting retrieved image quality is almost the same as that of full search method.

The detailed steps of the FIC using DWT are stated as follows:

1. Calculate *LHnr* and *HLnr* of all the range blocks and *LHnd* and *HLnd* of all the domain blocks.
2. $j = 0$.
3. $i = 0$.
4. Select the optimal Dihedral transformation T_k between r_j and d_i from the Table 3. Take T_k transformation to d_i according to the Table 2 to obtain the optimal transformed block $T_k(d_i)$.
5. Perform the similar match of r_j and $T_k(d_i)$. If the MSE value of the domain block d_i is smaller than that of the preceding optimal domain blocks, record its parameters as fractal code of r_j.

Table 3
The optimal Dihedral transformation relationship between the range block and domain block by using *LHn* and *HLn*.

Range	Domain							
	$LHnd > 0$ $HLnd > 0$ $\|LHnd\| > \|HLnd\|$	$LHnd > 0$ $HLnd < 0$ $\|LHnd\| > \|HLnd\|$	$LHnd < 0$ $HLnd > 0$ $\|LHnd\| > \|HLnd\|$	$LHnd < 0$ $HLnd < 0$ $\|LHnd\| > \|HLnd\|$	$LHnd > 0$ $HLnd > 0$ $\|LHnd\| < \|HLnd\|$	$LHnd < 0$ $HLnd > 0$ $\|LHnd\| < \|HLnd\|$	$LHnd > 0$ $HLnd < 0$ $\|LHnd\| < \|HLnd\|$	$LHnd < 0$ $HLnd < 0$ $\|LHnd\| < \|HLnd\|$
$LHnr > 0$ $HLnr > 0$ $\|LHnr\| > \|HLnr\|$	T_0	T_1	T_2	T_3	T_4	T_6	T_5	T_7
$LHnr > 0$ $HLnr < 0$ $\|LHnr\| > \|HLnr\|$	T_1	T_0	T_3	T_2	T_6	T_4	T_7	T_5
$LHnr < 0$ $HLnr > 0$ $\|LHnr\| > \|HLnr\|$	T_2	T_3	T_0	T_1	T_5	T_7	T_4	T_6
$LHnr < 0$ $HLnr < 0$ $\|LHnr\| > \|HLnr\|$	T_3	T_2	T_1	T_0	T_7	T_5	T_6	T_4
$LHnr > 0$ $HLnr > 0$ $\|LHnr\| < \|HLnr\|$	T_4	T_5	T_6	T_7	T_0	T_2	T_1	T_3
$LHnr < 0$ $HLnr > 0$ $\|LHnr\| < \|HLnr\|$	T_5	T_4	T_7	T_6	T_2	T_0	T_3	T_1
$LHnr > 0$ $HLnr < 0$ $\|LHnr\| < \|HLnr\|$	T_6	T_7	T_4	T_5	T_1	T_3	T_0	T_2
$LHnr < 0$ $HLnr < 0$ $\|LHnr\| < \|HLnr\|$	T_7	T_6	T_5	T_4	T_3	T_1	T_2	T_0

6. Increment i by one. If i is smaller than $(m-2n+1)^2$, go to step 4.
7. Increment j by one. If j is smaller than $(m/n)^2$, go to step 3. Otherwise stop.

In Step 4, r_j and d_i represent jth range block and ith domain block, respectively. $T_k(d_i)$ is the transformed block obtained by taking T_k transformation to d_i. In Step 6 and 7, the quantities $(m-2n+1)^2$ and $(m/n)^2$ are the size of domain pool and range pool, respectively, described in Section 2. Since rang block performs the similar match only with the optimal Dihedral transformed block, the number of the MSE computations of the method is reduced to $(m-2n+1)^2 \times (m/n)^2$.

4. Genetic algorithm based on DWT for fractal image compression

In Section 3, by using two discrete wavelet coefficients *LHn* and *HLn*, the FIC using DWT method in comparison to the full search method can reduce seven eighths number of the MSE computations. In this section, the two discrete wavelet coefficients *LHn* and *HLn* are embedded into the GA to speedup the fractal encoder further. The GA based on DWT will overcome the disadvantages of the traditional GA method described in Section 1 to settle the trade-off problem between retrieved quality and encoding speed.

In the proposed GA method, the chromosome is formed using the x-coordinate and y-coordinate of the domain block in the image. The reason is that, for each range block, the optimal Dihedral transformed block of the domain block represented by the chromosome can be determined directly from Table 3 according to their *LHn* and *HLn*. Since the optimal Dihedral transformation is obtained from the DWT coefficients, it is not necessary to include the Dihedral index in the GA chromosome. The short chromosome represents a smooth search space and easy to find a good solution. Furthermore, although the chromosome stands for a domain block, but its optimal Dihedral transformed block can be determined from Table 3 in advance. Hence only the optimal Dihedral transformed block for the chromosome is taken to do the similar match with the range block. The other seven Dihedral blocks will be ignored. Integrating with the statements above, the GA based on DWT method can improve indeed the trade-off problem between retrieved quality and encoding speed for the traditional GA method. The setup of the proposed GA method is summarized as follows:

(1) Chromosome formation: As stated in above, the chromosome $c_i, i = 1, 2, \ldots, M$ are constituted by x-coordinate and y-coordinate of the domain block in an image. Here, M is the size of the population.
(2) Fitness function: The similarity between the range block r and the domain block d is measured by their MSE. Since the smaller the MSE, the higher the similarity, the fitness value is defined as the inverse of MSE.
(3) Selection mechanism: Selection mechanism selects two parents from the mating pool to execute the crossover operation. In our proposed method, the ranking selection is adopted to select the good chromosomes to evolve and avoid precocity of the population.
(4) Crossover operation: Each pair of parents selected from the mating pool undergoes the crossover operation to generate the temporal offspring. The goal is to exploit the better candidate solution at the neighborhood of the parents. In our proposed method, the uniform crossover is used.
(5) Mutation operation: The mutation operations are performed on the temporal offspring to generate the chromosomes for the next generation. The goal is to carry out the global exploration in the whole search space in order to avoid pre-maturity.
(6) Stopping criterion: The stopping criterion is when a pre-set number of iterations L is reached, the GA evolution will be stopped.

(7) Elitism: The elitism preserves good chromosomes generated in each of the GA iteration and directly sends them into the next generation. The number of the selected good chromosomes is determined according to the user's requirement.

According to the above operational strategies, a GA based on DWT for FIC method is proposed. The detailed steps are given as follows.

Initially, set the population size M, the crossover mask, crossover rate, mutation mask, mutation rate, the number of iteration L.

1. Calculate *LHnr* and *HLnr* of all the range blocks and *LHnd* and *HLnd* of all the domain blocks.
2. $j = 0$.
3. Generate the initial chromosomes $c_i, i = 1, 2, \ldots, M$ randomly.
4. For all the chromosomes, select the optimal Dihedral transformation T_k between r_j and c_i from the Table 3. Take T_k transformation to c_i according to the Table 2 to obtain the optimal transformed block $T_k(c_i)$. Perform the similar match of r_j and $T_k(c_i)$.
5. Rank the chromosomes according to their fitness values.
6. If a pre-set number of iterations L is reached, record the fractal code of r_j and go to step 10. Otherwise, go to next step.
7. Select the parent chromosomes according to the select mechanism.
8. Perform the uniform crossover to generate the temporary offspring.
9. Perform the mutation operation on the temporary offspring to generate the chromosomes of the next generation. Go to step 4.
10. Let j add one. If j is equal to $(m/n)^2$, then stop, otherwise go to step 3.

Similar to the algorithm given in Section 3, the quantity $(m/n)^2$ is the number of the range block. Let $T_k(c_i)$ be the optimal transformed block obtained by taking T_k transformation on the block corresponding to the chromosome c_i. In each of the GA iteration, range block r_j does the similar match with the optimal transformed blocks of M chromosomes. Hence, under the condition of the elitism not considered, the number of the MSE computations for the proposed GA method is equal to $(m/n)^2 \times M \times L$ in total. Furthermore, if the elitism is considered, the number of the MSE computations can be reduced further.

5. Experimental results

The performances of the proposed methods are simulated and verified. The tested images are Lena, Pepper, F16, and Baboon, each of which is of size 256×256. The size of range and domain blocks for fractal coder is 8×8 and 16×16, respectively. The software simulation is implemented using Borland C++ Builder (BCB) v6.0 running on a Pentium 2.0 GHz Windows XP PC. The difference of image quality between the original image $f(i,j)$ and the retrieved image $g(i,j)$ is measured by Peak Signal to Noise Ratio (PSNR) defined as

$$\mathrm{PSNR} = 10 \times \log\left(\frac{255^2}{\mathrm{MSE}}\right),$$

where $\mathrm{MSE} = \frac{1}{m^2}\sum_{i,j=0}^{m-1}(f(i,j) - g(i,j))^2$ and m is the image size.

5.1. The comparison of FIC using DWT to full search method

The simulations of the FIC using DWT and full search methods are performed to compare their efficiency. For the FIC using DWT, the Haar wavelet is used. The range block and domain block are decomposed using DWT up to 3 levels and 4 levels, respectively, since their sizes are 8×8 and 16×16. Hence, in the Table 3, the *LHnr* and *HLnr* of range block shown in the first column are *LH3r* and *HL3r*, respectively, and the *LHnd* and *HLnd* of the domain block shown in the first row are *LH4d* and *HL4d*, respectively.

Table 4
The comparison of PSNR and the number of the MSE computations between the FIC using DWT and full search methods.

Image	Method	No. of MSE computations	PSNR (dB)
Lena	Full search	475,799,552	28.91
	FIC using DWT	59,474,944	28.55
Pepper	Full search	475,799,552	29.84
	FIC using DWT	59,474,944	29.47
F16	Full search	475,799,552	25.21
	FIC using DWT	59,474,944	24.70
Baboon	Full search	475,799,552	20.15
	FIC using DWT	59,474,944	20.04

Table 4 shows the comparison results between the FIC using DWT and full search methods at the PSNR and the number of the MSE computations. For the computational load of all the four tested images, the numbers of the MSE computations of the FIC using DWT and full search methods are 59,474,944 and 475,799,552, respectively. As discussed in Section 3, the FIC using DWT method indeed needs only one eighth number of the MSE computations. At the PSNR, the PSNR of the FIC using DWT and full search methods for the Lena is 28.55 dB and 28.91 dB, respectively. The difference of the retrieved image quality between the two methods is only 0.36 dB. Moreover, for the Pepper, F16, and Baboon, the decays of the retrieved image are also only 0.37 dB, 0.51 dB, and 0.11 dB, respectively. The quality of the retrieved image of the FIC using DWT is almost the same as that of the full search method.

5.2. The performance of the GA based on DWT for FIC

The Schema GA (SGA) method in [15] is used to compare with our GA method to demonstrate the performance improvement of our GA method. In the proposed GA method, the chromosome is composed of the x-coordinate and y-coordinate of an image. The crossover probability is 0.6 and the mutation probability is 0.05. In SGA method, the chromosome is formed by the x-coordinate and y-coordinate of an image and Dihedral index. The crossover probability is 0.6. The mutation probability $P_{m,b}$ of the superior clan and the mutation probability $P_{m,w}$ of the inferior clan are 0.05. The GA parameters for the two methods are the same except the chromosome formation. The number of the chromosome for the two

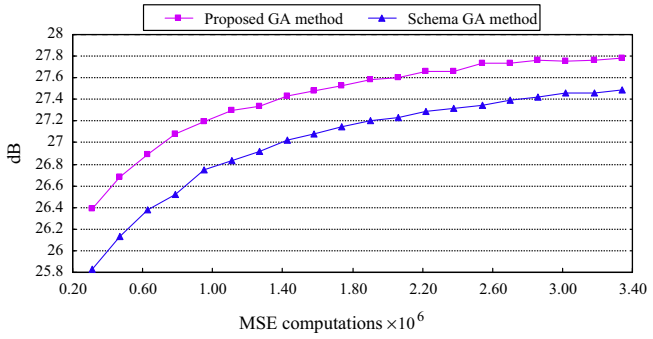

Fig. 3. The comparison of the PSNR versus the number of MSE computations for Lena.

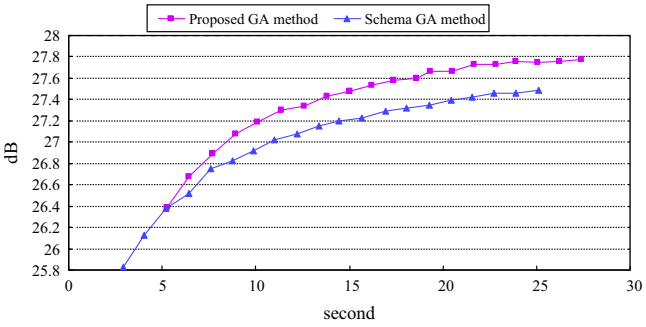

Fig. 4. The comparison of the PSNR versus the encoding time for Lena.

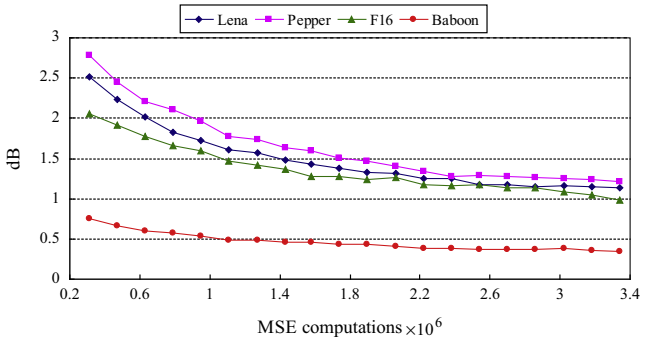

Fig. 5. Decayed PSNR of the retrieved images versus the number of the MSE computations.

methods is set to be 300. Furthermore, the data are obtained by performing 10 runs and averaging the results to achieve proper statistic comparison.

Fig. 3 shows the relation of the retrieved image quality versus the number of MSE computations for the Lena. The data points from left to right in Fig. 3 are acquired by executing the number of the iteration of the GA from 1 to 20 in order. The figure shows that, under each of the number of the fixed iteration of GA, the number of the MSE computations for the two methods are very closed, because their GA parameters are the same. In the beginning, the PSNRs of the proposed GA and SGA methods are 26.38 dB and 25.83 dB, respectively. After the 4th iteration, the PSNR of our method is beyond 27 dB already. The result exhibits one important feature of the proposed GA method, i.e., we can obtain acceptable retrieved image quality using only a small number of GA iterations. Finally, at the 20th iteration, the PSNRs of the proposed GA and SGA methods are 27.78 dB and 27.49 dB, respectively. As simulated, the proposed GA method, compared to SGA method, reduces about 0.29 to 0.47 dB decay at retrieved image under the same number of the MSE computations.

The retrieved image quality versus the encoding time is depicted in Fig. 4. Similarly to Fig. 3, the data points from left to right in Fig. 4 represents the number of the iterations of GA evolution from 1 to 20. The curve of proposed GA method compared to the curve of SGA method is shifted slightly toward the right, since the proposed GA method must expend a little time on the wavelet decomposition to find *LHn* and *HLn* of all the blocks. Further to observe, although the encoding time of the proposed GA method executing 1 iteration is almost equal to that of the SGA method executing 3 iterations, their retrieved image qualities are almost the same. As the encoding time increases, the difference in the retrieved image quality is enlarged gradually. The retrieved quality of proposed GA method is still better than that of the SGA method at the same encoding time.

Fig. 5 shows the decayed PSNR of the retrieved image versus the number of MSE computations for the four tested images using the proposed GA method. The decayed PSNR is obtained by using the full search method as the benchmark for the comparison. The figure shows that the decay speed is fast at the beginning. As the number of MSE computations is more than two million, the quality decay slows down and maintains in an acceptable range. The phenomenon is independent of the type of images.

Table 5 compares the performance difference of GA based on DWT, Duh's method [12], Schema GA, FIC using DWT, and full search methods for Lena of size 256×256. For the FIC using DWT, the encoding time is 425.08 s and the PSNR is 28.55 dB. The decay of PSNR is only 0.36 dB. Hence its retrieved image quality is almost the same as that of full search method. As observed, the speedup ratio is 6.44 in the encoding time while the amount of MSE computation is reduced 8 times. This is because the proposed method requires some overhead to calculate the wavelet coefficients and the optimal Dihedral index. For the GA based on DWT, the encoding time and the PSNR are 27.33 s and 27.78 dB, respectively, in which the number of iterations of GA is 20. Under the same number of iterations, the decay of PSNR, compared to SGA method, is improved 0.29 dB with the penalty of outrunning 2.22 s encoding time. Moreover, in comparison to Duh's method, the decay of PSNR also is reduced from 1.32 dB to 11.3 dB under the roughly same number of the MSE computations. However, the encoding time is slightly increased due to the overhead of GA operations. Finally, compared to the full search method, the speedup ratio and reduced ratio are 100.09 and 142.41, respectively. The loss is 1.13 dB decay at the retrieved image quality. Similarly, because the GA based on DWT method requires some overhead to calculate the wavelet coefficients, the optimal Dihedral index, and the GA evolutionary, the value of speedup ratio is also smaller than that of the reduced ratio. The reason is the same as FIC using DWT.

Fig. 6 shows the decoded images under full search, FIC using DWT, and GA based on DWT methods. The parameters are shown in Table 5. As the visual effect shown in the figures, under the speedup ratio of 6.44 times, the retrieved image of the FIC using

Table 5
The performance of GA based on DWT, Duh's method, Schema GA, FIC using DWT and full search methods; the tested image is Lena of size 256×256.

Compression method	Full search	FIC using DWT	Schema GA	Duh's method	GA based on DWT
No. of iterations	–	–	20		20
Category number				142	
PSNR (dB)	28.91	28.55	27.49	27.59	27.78
Decayed PSNR (dB)	–	0.36	1.42	1.32	1.13
Time (s)	2735.58	425.08	25.11	22.25	27.33
Speedup ratio	1.00	6.44	108.94	122.95	100.09
No. of MSE computations	475,799,552	59,474,944	3,341,229	3,350,736	3,341,109
Reduced ratio	1.00	8.00	142.40	142.00	142.41

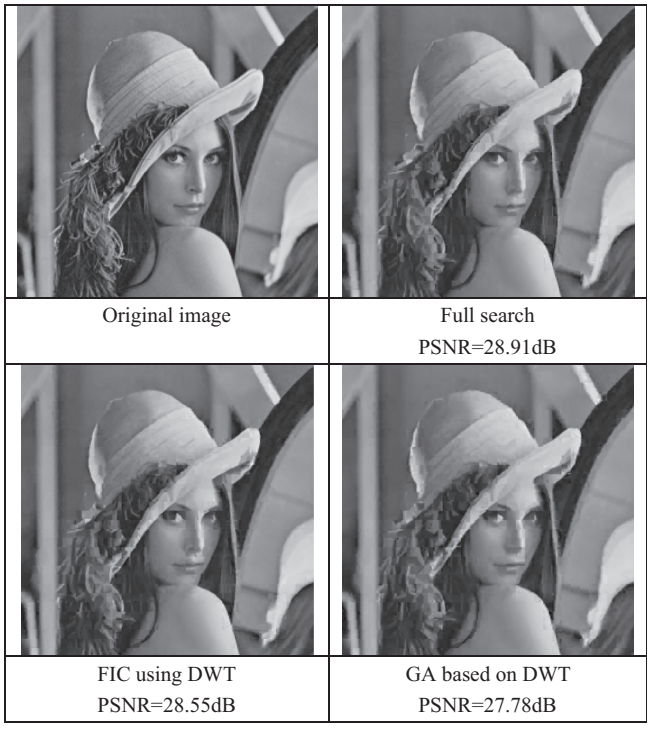

Fig. 6. Retrieved images under full search, FIC using DWT, and GA based on DWT methods.

DWT method is the same as that of the full search method. Moreover, embedding the DWT into the GA, the speedup ratio is increased to 100 times. Nevertheless the retrieved image quality is still acceptable relatively.

6. Conclusion

In this paper, two fractal encode algorithms have been proposed to overcome the problem of the time-consuming drawback for the fractal encoder. First, a FIC using DWT is proposed to ignore unnecessary MSE computations produced by Dihedral transformations. For each range block, two discrete wavelet coefficients: *LHn* and *HLn* were used to determine the fittest Dihedral index of the domain block. The range block does the similar match only with the optimal transformed block of the domain block and the other seven transformed blocks are discarded. Compared to full search method, the method achieves 6.44 times speedup ratio with the almost same retrieved image quality. Second, embedding the DWT technique into the genetic algorithm, a GA based on DWT is implemented to overcome the trade-off problem between the quality and speed for the traditional GA. The proposed GA method has the advantages of the fast speed of evolution and the less number of MSE computations, since the optimal Dihedral index was determined to shorten the length of the chromosome effectively. Compared to SGA method, the proposed GA method reduces about 0.29 to 0.47 dB decay at retrieved image under the same number of the MSE computations. Moreover, at the encoding speed, the proposed GA method is 100 times faster than the full search method with the penalty of 1.13 dB decay at the retrieved image quality.

References

[1] M.F. Barnsley, S. Demko, Iterated function systems and the global construction of fractals, Proc. Roy. Soc. A399 (1985) 243–275.
[2] A.E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Trans. Image Process. 1 (1) (1992) 18–30.
[3] H.O. Peitgen, J.M. Henriques, L.F. Penedo, Fractals in the Fundamental and Applied Sciences, Elsevier Science Publishing Company Inc., New York, 1991.
[4] J. Crilly, R.A. Earnshaw, H. Jones, Fractals and Chaos, Springer-Verlag, New York, 1991.
[5] T.K. Truong, C.M. Kung, J.H. Jeng, M.L. Hsieh, Fast fractal image compression using spatial correlation, Chaos Soliton. Fract. 22 (5) (2004) 1071–1076.
[6] S. Furao, O. Hasegawa, A fast no search fractal image coding method, Signal Process.: Image Commun. 19 (5) (2004) 393–404.
[7] D.J. Duh, J.H. Jeng, S.Y. Chen, DCT based simple classification scheme for fractal image compression, Image Vision Comput. 23 (2005) 1115–1121.
[8] X.Y. Wang, F.P. Li, Z.F. Chen, An improved fractal image coding method, Fractals 17 (4) (2009) 451–457.
[9] X.Y. Wang, Y.X. Wang, J.J. Yun, An improved no-search fractal image coding method based on a fitting plane, Image Vision Comput. 28 (8) (2010) 1303–1308.
[10] Z. Wang, D. Zhang, Y.L. Yu, Hybrid image coding based on partial fractal mapping, Signal Process.: Image Commun. 15 (2000) 767–779.
[11] T. Kova'cs, A fast classification based method for fractal image encoding, Image Vision Comput. 26 (2008) 1129–1136.
[12] D.J. Duh, J.H. Jeng, S.Y. Chen, Speed quality control for fractal image compression, Imag. Sci. J. 56 (2008) 79–90.
[13] X.Y. Wang, Y. Lang, A fast fractal encoding method based on fractal dimension, Fractals 17 (4) (2009) 459–465.
[14] X.Y. Wang, Y.X. Wang, J.J. Yun, An improved fast fractal image compression using spatial texture correlation, Chin. Phys. B 20 (10) (2011) 104202.
[15] K.L. Chung, C.H. Hsu, Novel prediction- and subblock-based algorithm for fractal image compression, Chaos Soliton. Fract. 29 (2006) 215–222.
[16] X.Y. Wang, S.G. Wang, An improved no-search fractal image coding method based on a modified gray-level transform, Comput. Graph. 32 (2008) 445–450.
[17] M.S. Wu, W.C. Teng, J.H. Jeng, J.G. Hsieh, Spatial correlation genetic algorithm for fractal image compression, Chaos Soliton. Fract. 28 (2006) 497–510.
[18] Y. Zhang, X.Y. Wang, Fractal compression coding based on wavelet transform with diamond search, Nonlinear Anal.: Real World Appl. 13 (2012) 106–112.
[19] M.S. Wu, J.H. Jeng, J.G. Hsieh, Schema genetic algorithm for fractal image compression, Eng. Appl. Artif. Intell. 20 (2007) 531–538.
[20] C.C. Tseng, J.G. Hsieh, J.H. Jeng, Fractal image compression using visual-based particle swarm optimization, Image Vision Comput. 26 (2008) 1154–1162.
[21] X.Y. Wang, F.P. Li, S.G. Wang, Fractal image compression based on spatial correlation and hybrid genetic algorithm, J. Visual Commun. Image Represent. 20 (8) (2009) 505–510.
[22] A. Muruganandham, R.S.D. Wahida Banu, Adaptive fractal image compression using PSO, Proc. Comput. Sci. 2 (2010) 338–344.
[23] M.S. Wu, Y.L. Lin, Genetic algorithm with a hybrid select mechanism for fractal image compression, Digital Signal Process. 20 (4) (2010) 1150–1161.
[24] Y. Fisher, Fractal Image Compression: Theory and Application, Springer-Verlag, New York, 1994.