Fractal Image Compression Based on Spatial Correlation And Chaotic Particle Swarm Optimization

Gohar Vahdati, Mahdi Yaghoobi
Computer Engineering Department
Islamic Azad University, Mashhad Branch
Mashhad, Iran
1Elham62.vahdati@gmail.com, 2yaghobi@mshdiau.ac.ir

Mohammad Reza Akbarzadeh- Totonchi
Electrical Engineering Department
Ferdowsi University of Mashhad
Mashhad, Iran
akbarzadeh@ieee.org

Abstract— Fractal image compression explores the self-similarity property of a natural image and utilizes the partitioned iterated function system (PIFS) to encode it. This technique is of great interest both in theory and application. However, it is time-consuming in the encoding process and such drawback renders it impractical for real time applications. The time is mainly spent on the search for the best-match block in a large domain pool. In order to solve the high complexity of the conventional encoding scheme for fractal image compression, a spatial correlation chaotic particle swarm optimization (SC-CPSO), based on the characteristics of fractal and partitioned iterated function system (PIFS) is proposed in this paper. There are two stages for the algorithm: (1) Make use of spatial correlation in images for both range and domain pool to exploit local optima. (2) Adopt chaotic PSO (CPSO) to explore the global optima if the local optima are not satisfied. Experiment results show that the algorithm convergent rapidly. At the premise of good quality of the reconstructed image, the algorithm saved the encoding time and obtained high compression ratio.

Keywords- Fractal image compression; chaotic particle swarm optimization; encoding time; spatial correlation.

I. INTRODUCTION

The idea of the fractal image compression (FIC) is based on the assumption that the image redundancies can be efficiently exploited by means of block self-affine transformations [1,2]. In 1988, Barnsley [3] proposed the idea of fractal image compression for the first time. The first practical fractal image compression scheme was introduced in 1992 by Jacquin [4]. One of the main disadvantages using exhaustive search strategy is the low encoding speed. Therefore, improving the encoding speed is an interesting research topic for FIC. So far, some improved approaches have been presented. Fisher and Wang et al. proposed their classification methods [5-6] based on the feature of the domain blocks, respectively. Truong et al.[7] proposed a kind of neighborhood matching method based on spatial correlation which makes use of the information of matched range blocks and effectively reduced the encoding time. Some other researchers have combined fractal with other algorithms such as ant colony optimization [8], genetic algorithm[9], wavelet [10], etc. A schema genetic algorithm for fractal image compression is proposed in [11] to find the best self similarity in fractal image compression. Wu et al. [12] proposed a Spatial Correlation Genetic Algorithm (SC-GA), which speeded up the encoding time and increased compression ratio. PSO is an optimization algorithm having origins from evolutionary computation together with the social psychology principle [13-15]. The mechanism of PSO algorithm generating the optimal or near-optimal solutions is a stochastic process. The formulae of the traditional PSO algorithm are simpler than those of the SA and the ACO algorithm, which means that the PSO algorithm can be more conveniently implemented. The population size of the PSO algorithm is smaller than those of the SA and GA algorithm, so the initialization of the populations is simpler using the PSO algorithm than that of the other intelligent optimization algorithms. Although the traditional PSO algorithm has many advantages to resolve the optimization problems, the performance of it, such as the premature convergence and the stochastic stagnation, is heavily impacted by the principle and the parameters of the algorithm. The research [16] indicates that the traditional PSO algorithm will smoothly slip into the local near-optimal solutions when the optimization problem is relatively complex and it cannot jump over the obstruction. The adverse situations will be lowered after the chaos method is combined with the traditional PSO algorithm. Chaos method [17-18] has the particular characteristics, such as the randomness and ergodicity, which can enhance the diversity of the particles and actuate the particles to move out from the local near-optimal solutions. The premature convergence of the traditional PSO algorithm is weakened and the convergence of it is also accelerated. In this paper, a fractal image compression algorithm based on spatial correlation and chaotic particle swarm optimization algorithm is proposed. Results show that proposed algorithm reduced the coding time and retained high compression ratio under the premise of good image quality.

II. FULL SEARCH FRACTAL IMAGE COMPRESSION

The fundamental idea of fractal image compression is the Iteration Function System (IFS) in which the governing theorems are the Contractive Mapping Fixed-Point Theorem and the Collage Theorem [5]. For a given gray level image of size N x N, let the range pool R be the set of the (N/L)²
non-overlapping blocks of size L×L, which is the size of encoding unit. Let the contractivity of the fractal coding be a fixed quantity of 2. Thus, the domain pool makes up the set of \((N-2L+1)^2\) overlapping blocks of size \((2L×2L)\). For the case of 256×256 image with 8×8 coding size, the range pool contains 1024 blocks of size 8×8 and the domain pool contains 58081 blocks of size 16×16. For each range block \(v\) in \(R\), one searches in the domain pool \(D\) to find the best match, i.e., the most similar domain block. The parameters describing this fractal affine transformation form the fractal compression code of \(v\). At each search entry, the domain block is first down-sampled to 8×8 and denoted by \(u\). Let the set of down-sampled domain blocks be denoted by \(D\). The down-sampled block is transformed subject to the eight transformations in the Dihedral on the pixel positions [15].

The eight transformed blocks are denoted by \(u_k\), \(k=0,1, ..., 7\). In fractal coding, it is also allowed a contrast scaling \(p\) and a brightness offset \(q\) on the transformed blocks [15]. As \(u\) runs over all of the 58081 domain blocks in \(D\) to find the best match, the position of the domain block (the terms \(t_x\) and \(t_y\)), the contrast scaling \(p\), the brightness offset \(q\) and the orientation \(d\), is found for the given range block \(v\). In practice, \(t_x\), \(t_y\), \(d\), \(p\), and \(q\) can be encoded using 8, 8, 3, 5, and 7 bits, respectively, which are regarded as the compression code of \(v\). Finally, as \(v\) runs over all of the 1024 range blocks in \(R\), the encoding process is completed.

III. Chaotic Particle Swarm Optimization

A. Particle Swarm Optimization

PSO is a population-based algorithm for searching global optimum. The original idea of PSO is to simulate a simplified social behavior. It ties to artificial life, like bird flocking or fish schooling, and has some common features of evolutionary computation such as fitness evaluation. The adjustment toward the best individual experience (PBEST) and the best social experience (GBEST) is conceptually similar to the crossover operation of the GA. However, it is unlike a GA in that each potential solution, called particle, is “flying” through hyperspace with a velocity. Moreover, the particles and the swarm have memory, which does not exist in the population of the GA [13-14]. PSO is initialized with a swarm including \(N\) random particles. Each particle is treated as a point in a \(D\)-dimensional space. The \(i\)-th particle is represented as \(x_i = (x_{i1}, x_{i2}, ..., x_{iD})\), \(x_{i1}\) is limited in the range \([a_i, b_i]\). The best previous position of the \(i\)-th particle (PBEST), is represented as \(p_i = (p_{i1}, p_{i2}, ..., p_{iD})\). The best particle among all the particles in the population (GBEST), is represented by \(p_g = (p_{g1}, p_{g2}, ..., p_{gD})\). The velocity of particle \(i\) is represented as \(v_i = (v_{i1}, v_{i2}, ..., v_{iD})\) [17]. After finding the aforementioned two best values, the particle updates its velocity and position according to the following equations:

\[
\begin{align*}
\dot{v}_{id} &= v_{id} + c_1 r_1(p_{id} - x_{id}) + c_2 r_2(p_{gd} - x_{id}) \\
x_{id} &= x_{id} + v_{id}
\end{align*}
\]

where \(d\) is the \(d\)-th dimension of a particle, \(c_1\) and \(c_2\) are two positive constants called learning factors, \(r_1\) and \(r_2\) are random numbers in the range of \([0,1]\).

B. Particle Swarm Optimization Based on Chaos Searching

Chaos is characterized as ergodicity, randomicity and regularity [17-18]. Because chaos queues can experience all the states in a specific area without repeat, chaotic search becomes a novel tool used as an optimizer. Here, Logistic equation is employed to obtain chaos queues, which is represented as follows [17]:

\[
z_{n+1} = \mu z_n (1 - z_n) \quad n = 0, 1, 2, ...
\]

Where \(\mu\) is the control parameter, suppose that \(0 \leq z_0 \leq 1\), when \(\mu = 4\), the system of (3) has been proved to be entirely chaotic. The basic ideas of CPSO are described as follows:

Generate the chaotic queues based on the optimal position searched by all particles until now, and then replace the position of one particle of the swarm with the best position of the chaos queues. The searching algorithm using chaotic queues can avoid the search being trapped in local optimum and can help to search the optimum quickly [17].

The detailed design of chaotic pso (CPSO) is summarized as follows:

Step 1: Randomly initialize the position of \(M\) particles.

Step 2: Randomly initialize the velocity of \(M\) particles.

Step 3: For all the particles of the swarm, take the following steps:

1. Using the global best and the individual best of each particle, each particle’s velocity and position are updated according to (1) and (2).

2. Evaluate the fitness of each particle and compare the evaluated fitness value of each particle to its individual best \(p_i\). If current value is better than \(p_i\), then update \(p_i\) as current position.

3. If current value is better than the global best \(p_g\), then update \(p_g\) as current position.

Step 4: Optimize \(p_g\) by chaos search. Firstly, scale \(P_{gi}\) \((i=1,2, ..., l)\) into \([0,1]\) according to \(z_i = (p_{gi} - a_i)/(b_i - a_i)\), \((i=1,2, ..., l)\), and generate chaos queues \(z_i^{(m)}\) \((m=1,2, ...)\) by iteration of Logistic equation, then, transfer the chaotic queues into the optimization variable \(p_{g(m)}\) \((m=1,2, ...)\) according to following equation: \(p_{g(m)}^{(m)} = a_i + (b_i - a_i)z_i^{(m)}\), upon that the solution set is obtained:

\[
P_{g(m)} = (p_{g1}^{(m)}, p_{g2}^{(m)}, ..., p_{gD}^{(m)}) \quad (m=1,2, ...) \]

Compute the fitness value of each feasible solution \(p_{g(m)}^{(m)}\) \((m=1,2, ...)\) in the problem space during chaotic search, and get the best solution \(p^*\).

Step 5: Replace the position of one particle selected randomly from the swarm with \(p^*\).

Step 6: If one of the stopping criteria is satisfied, then stop, we can get the global optimum that are the solutions of the problem. Otherwise, loop to Step 3.
IV. FRACTAL IMAGE COMPRESSION BASED ON SPATIAL CORRELATION AND CPSO (SC-CPSO)

The proposed method is implemented in two stages. The first stage makes full use of spatial correlations in images to exploit local optima. It can reduce the searching space of the similar matching domain pool, and shorten the optimal searching time. The second stage is operated on the whole image to explore more adequate similarities if the local optima are not satisfied. Let \( r_i \) be the range block to be encoded, \( 0 \leq i < 1024 \). Denote the neighbor range blocks of \( r_i \) as depicted in Figure 1, by \( r^{H}, r^{V}, r^{D} \) and \( r^{D'} \) which have been encoded. Assume \( d^{H(1)}, d^{V(1)}, d^{D(1)} \) and \( d^{D'(1)} \) are the corresponding matched domain blocks, respectively. Now, one will restrict the searching space of \( r_i \) to \( d^{H(1)}, d^{V(1)}, d^{D(1)}, d^{D'(1)} \) including some domain blocks in the relative directions. For example, \( d^{H(1)} \) is the mapped domain block of \( r_i \) which is in the horizontal direction of \( r_i \). Thus one expands the searching space in the horizontal direction (\( S^H \)) to \( d^{H(0)}, d^{H(1)}, d^{H(2)} \) and \( d^{H(3)} \) as depicted in Figure 1. Similarly, \( d^{V(1)}, d^{D(1)} \) and \( d^{D'(1)} \) are expanded to \( S^V, S^D, S^{D'} \) according to their corresponding directions.

![Figure 1](image_url)

Figure 1. The searching space of the current range block in the first stage.

Thus, the searching space of \( r_i \) is limited to

\[
S = \{S^H, S^V, S^D, S^{D'}\}
\]

(4) Perform the second stage of the proposed algorithm (CPSO), on \( r_i \). Let \( d_{k}^{j} \) be the best-matched block in this stage. record the fractal code.

(5) Let \( i = i + 1 \), If \( i \) is equal to 1024, then stop, otherwise go to step (2).

In step (2), since the range blocks located at the first row, the first column, the last row, and the last column do not have the neighbors as shown in Figure 1, the first stage of the proposed algorithm is not performed on these blocks[19]. if the fractal codes come from the first stage, the range block \( r_i \) is called “hit” block, which indicates that the local optima can satisfy the demand. For a hit block, fewer bits are required to record the offset of the domain block instead of the 16-bit absolute position. This will improve the compression ratio. Let \( N_R \) and \( N_H \) denote the number of range blocks and hit blocks, respectively. For hit blocks, \( 2 + B_w \) bits are required to record the relative positions, where \( B_w \) denotes the number of bits to represent the expansions. Let \( B_k, B_p \) and \( B_q \) denote the number of bits required to represent the orientation, the contrast and the brightness, respectively. Then the bit rate (bit per pixel) can be computed directly in terms of the number of hit blocks as

\[
\text{bpp} = \frac{N_R}{N_T} \left(1 + 2 + B_{w} + B_{k} + B_{p} + B_{q}\right)
\]

where \( N_T \) is the total number of pixels in the image[7]. Note that one bit is required to indicate if the block is a hit block. As discussed in Section II, the parameters \( t_x, t_y, d, p, q \) constitute the fractal code. In the proposed method, we encode the particle as \( (t_x, t_y) \), which is the position of the domain block. At each search entry, all of the eight Dihedral transformations are performed and the best index \( d \) can be obtained. The fitness value of a particle is defined as the minus of the minimal MSE produced from eight Dihedral transformations, i.e., \(-\text{MSE}((p_d u_k + q_d) v)\) [15]. When the stopping criterion is satisfied, the final \( p_g \) with the corresponding \( d, p, q \) is the fractal code of the given range block \( v \). The steps of encoding a range block using CPSO are summarized as follows:

1. Initialize the parameters of PSO.
2. For each particle \((t_x, t_y)\), fetch the domain block at \((t_x, t_y)\) in the image. Sub-sample the block and denote it by \( u \).
3. For each Dihedral transformation, compute \( u_k \), \( k = 0, \ldots, 7 \). Calculate the contrast scaling \( p_d \) and brightness offset \( q_d \). Find the fitness of the particle corresponding to the best parameter \( d \) [15].
4. Update the PBEST\((p_k)\) and the GBEST\((p_g)\) if required. The corresponding fractal codes are also updated accordingly.
5. Optimize \( p_g \) by chaotic search.
6. Replace the position of one particle selected randomly from the swarm with \( p^* \).
7. If stopping criterion is satisfied, then stop.
8. For each particle, update the velocity and the position according (1) and (2). Go to step 2.

V. EXPERIMENTAL RESULTS

The proposed algorithm is examined on images Lena, Pepper and Baboon with the size of 256×256 and gray scale. The size of range blocks is considered as 8×8 and the size of domain blocks is considered as 16×16. In order to compare the quality of the search results, the MSE threshold (T) values are set to be 100, 300 and 80, for the images Lena, Baboon, and Pepper, respectively. In our experiments, the population size of the swarm is set to be 35 and the maximum number of iterations is set to be 33. The coefficients of PSO are set heuristically as C1=1.3 and C2=1.4. Table I, shows the experimental results on the Proposed method, Full Search method and Traditional GA method. Table II, shows some comparative results for both our method and SC-GA method[12],[19].

TABLE I. THE COMPARISON OF PROPOSED ALGORITHM(SC-CPSO) AND TRADITIONAL GA METHOD TOGETHER WITH FULL SEARCH METHOD

<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>PSNR</th>
<th>Speed-up rate</th>
<th>bpp</th>
<th>Hit block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Full Search</td>
<td>28.91</td>
<td>1</td>
<td>0.484</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>27.23</td>
<td>61.47</td>
<td>0.393</td>
<td>582</td>
</tr>
<tr>
<td></td>
<td>Traditional GA</td>
<td>26.23</td>
<td>54.35</td>
<td>0.484</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Full Search</td>
<td>29.84</td>
<td>1</td>
<td>0.484</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>28.32</td>
<td>56.16</td>
<td>0.387</td>
<td>613</td>
</tr>
<tr>
<td></td>
<td>Traditional GA</td>
<td>27.10</td>
<td>53.99</td>
<td>0.484</td>
<td>-</td>
</tr>
<tr>
<td>Pepper</td>
<td>Full Search</td>
<td>20.15</td>
<td>1</td>
<td>0.484</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>19.69</td>
<td>68.97</td>
<td>0.4449</td>
<td>301</td>
</tr>
<tr>
<td>Baboon</td>
<td>Full Search</td>
<td>19.47</td>
<td>53.62</td>
<td>0.484</td>
<td>-</td>
</tr>
</tbody>
</table>

TABLE II. THE COMPARISON OF PROPOSED METHOD WITH SC-GA METHOD

<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>decay of image quality</th>
<th>Hit blocks</th>
<th>Speed-up rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>SC-GA</td>
<td>1.69</td>
<td>576</td>
<td>24.27</td>
</tr>
<tr>
<td></td>
<td>SC-CPSO</td>
<td>1.68</td>
<td>582</td>
<td>0.3934</td>
</tr>
<tr>
<td>Pepper</td>
<td>SC-GA</td>
<td>1.09</td>
<td>599</td>
<td>22.86</td>
</tr>
<tr>
<td></td>
<td>SC-CPSO</td>
<td>1.52</td>
<td>613</td>
<td>0.3878</td>
</tr>
<tr>
<td>Baboon</td>
<td>SC-GA</td>
<td>0.52</td>
<td>197</td>
<td>19.72</td>
</tr>
<tr>
<td></td>
<td>SC-CPSO</td>
<td>0.46</td>
<td>301</td>
<td>0.4449</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

In this paper, a fractal image compression method based on spatial correlation and chaotic particle swarm optimization is proposed. There are two stages for the algorithm. The first stage exploits local optima by making use of the spatial correlation between neighboring blocks. If the local optima are not satisfied, the second stage of the algorithm is carried out in order to explore further similarities from the whole image. Since the searching space in the first stage is much smaller, so the coding time is reduced. Such a method can speed up the encoder and also preserve the image quality. Moreover, the compression ratio can also be improved since only relative positions are recorded in the first stage of the algorithm.

REFERENCES