Journal of College of Education for pure sciences(JCEPS)
Web Site: http://eps.utq.edu.iq/ Email: eps_tqr@yahoo.com
Volume 7, Number 2, May 2017

Fractal Image Compression using Quantum PSO

Firas Sabar Miften[(1*)], Hazeem B. Taher[(1)], Kadhim Ajeel Obeaband[(2)]

(1) Computer Science Dept., Faculty of Education for Pure Sciences, University of Thi-Qar, Iraq

(2) Physics Dept., Faculty of Sciences, University of Thi-Qar, Iraq

[*]firassebar@yahoo.com

Abstract:

The self-similarity property is used by Fractal Image Compression (FIC) of a natural image.FIC performs tow processes encoded and decoded. The encoding time is very large for most existing algorithms to obtains the coded image.A new particle swarm optimization (PSO) technique forfast fractal encoding is proposed by many types of researchto reduce the encoding time. In this paper, Optimizing PSO by Quantum to reduce the encoding time. Quantum particle swarm optimization (QPSO) technique can speed up the fractal encoder and preserve the image.

Keywords: Fractal image compression; quantum particle swarmoptimization; particle swarmoptimization;encoding time.

Journal of College of Education for pure sciences(JCEPS)
Web Site: http://eps.utq.edu.iq/ Email: eps_tqr@yahoo.com
Volume 7, Number 2, May 2017

ضغط الصور باستخدام الفراكتل وPSO الكمية

فراس صبار[*1] حازم باقر[2] كاظم عجيل[3]

(1) قسم علوم الحاسبات، كلية التربية للعلوم الصرفة، جامعة ذي قار، العراق

(2) قسم الفيزياء،كلية العلوم،جامعة ذي قار،العراق

*firassebar@yahoo.com

الخلاصة:

استخدمت طريقة ضغط الصور الكسورية خاصية التشابه الذاتي لضغط الصور. الطريقة تتم بمرحلتين الترميز وفك الترميز حيث مرحلة الترميز تتطلب وقت طويل جدا للحصول على الصورة المضغوطة. لذلك اقترحت طريقة جديدة باستخدام **PSO** لتسريع من قبل عدة باحثين لتقليل وقت الترميز. في هذا البحث استخدمه**PSO** مع الكوانتم لتقليل الوقت. تقنية **QPSO** تستطيع تسريع الترميز مع الحفاظ على معالم الصورة

1- Introduction

The idea of fractal image compression (FIC) was originally introduced by Barnsley[1] and the first practical FIC scheme was realized by Jacquin in 1992 [2]. it is based on the partitioned iteration function system (PIFS) which utilized the self-similarity property in the image to achieve the purpose of compression [3]. The encoding process of the fractal image compression is time-consuming. The reason is thatmost of the encoding time is spent on a lot of computations of thesimilarity measure. Hence one of the main research direction for fractal image compression is focused on how to reduce the encoding time. In the past, FIC with some classification methods were usually adopted, in which ranges and domains were classified in the pre-processing step. At each search entry, only domains with a similar class were examined[4-6]. Domain pool reduction is another technique to reduce the encoding time. A common method was to restrict the search space to those blocks whose spatial location was near the range location [7]. Wavelet transform is used to decompose the original image to various frequency sub- bands in which the attributes can be extracted from the wavelet coefficients belonging to different sub-bands. The distribution of wavelet coefficients can be used in context- based multiscale classification of document image[8]. The fast and efficient algorithm [9] was applied to triangular mesh to approximate surface data using wavelet transform coefficients. It directly determined local area complexity in an image and divides square cells depending on complexity. In [10], the authors implemented a hybrid image classification method combining wavelet transform, rough set approach, and artificial neural.

The particle swarm optimization (PSO) is focused gradually, which is one of thecommonly used evolutionary algorithms[11-15]. PSO is an optimization algorithm havingorigins from evolutionary computation together with social psychology principle. Essentially, PSO is dependent on stochastic processes and also uses the concept of fitness similar to the negative of the cost of Genetic Algorithm (GA).

A quantum-inspired version of the PSOalgorithm (QPSO) was proposed veryrecently [16]. The QPSO algorithm permits all particles to have a quantum behaviorinstead of the classical Newtonian dynamics that was assumed so far in all versions of the PSO. Thus, instead of the Newtonian random walk, some sort of “quantum motion” is imposed in the search process. When the QPSO is tested against a set of benchmarking functions, it demonstrated superior performance as compared to the classical PSO but under the condition of large population sizes. One of the most attractive features of the new algorithm is the reduced number of control parameters. Strictly speaking, there is only one parameter required Tobe tuned in the QPSO.

2- The Classical PSO

It will be very instructive to review first the basics of the PSO method in order tointroduce the quantum version. In the Standard PSO model, each individual is treated as a volume-less particle in theD-dimensional space, with the position and velocity of *ith* particle represented as

$$V_i(t+1) = w \times V_i(t) + c_1 \times random \times (P_i - X_i(t)) + c_2 \times random \times \left(P_g - X_i(t)\right)(1)$$

$$X_{i(t+1)} = X_i(t) + V_i(t+1)(2)$$

where

- $c_1 and c_2$ are positive constant.
- random is arandom number in the range of [0,1].
- Parameter w is the inertia weight introduced to accelerate the convergence speed of the PSO.
- The vector $P_i = (P_{i1}, P_{i2}, P_{i3}, \dots \dots . P_{iD})$is the best previous position (the position giving the best fitness value) of particle $i$called pbest, and the vector $P_g = (P_{g1}, P_{g2}, P_{g3}, \dots \dots . P_{gD})$ is the position of the best particle among all the particles in the population and called gbest.

The steps involved here is the population size is first determined, and the velocity and position of each particle are initialized. Each particle moves according to [17], and the fitness is then calculated. Meanwhile, the best positions of each swarm and particles are recorded. Finally, as the stopping criterion is satisfied, the best position of the swarm is the final solution. The main steps are given as follows:

1. Set the swarm size. Initialize the velocity and the position of each particle randomly.
2. For each i, evaluate the fitness value of x_i and update the individual best position P_i , if better fitness is found.
3. Find the new best position of the whole swarm. Update the swarm best position x_i. if the fitness of the new best position is better than that of the previous swarm.
4. If the stopping criterion is satisfied, then stop.
5. For each particle, update the position and the velocity according (1)and (2). Go to step2.

3-Quantum Particle Swarm Optimization

The QPSO algorithm allows all particles to move under quantum-mechanical rules rather than the classical Newtonian random motion. In the classical environment, all bees are flying toward the optimum location. The particles are then attractedto this location through the optimization process. Such attraction leads to the global optimum. From equation (1) it is easy to see that is nothing but a random average of the global and local bests of the particles of the swarm. In quantum mechanics, the governing equation is the general time-dependent Schrödinger equation.

In Quantum-behaved Particle Swarm Optimization (QPSO), the particle moves according to the following equation:

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \quad \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \quad \cdots, \quad \frac{1}{M}\sum_{i=1}^{M} P_{id}\right) \tag{3}$$

$$p_{id} = \varphi * P_{id} + (1-\varphi) * P_{gd}, \quad \varphi = rand\,() \tag{4}$$

$$X_{id} = p_{id} \pm \alpha * |mbest_d - X_{id}| * \ln(1/u), \quad u = Rand() \tag{5}$$

Where

- *mbest* is the mean best position among the particles.
- P_{id}, a stochastic point between id P_{id} and P_{gd}, is the local attractor on the *dth* dimension of the *ith* particle.
- Φ and u are a random umber distributed uniformly on [0,1].
- Α is a parameter of QPSO that is called Contraction-Expansion Coefficient.

The Quantum-behaved Particle Swarm Optimization (QPSO) algorithm is described as follows.

1. Initialize an array of particles with arandom position inside the problem space.
2. Determine the mean best position among the particles by Eq(3)
3. Evaluate the desired objective function (for example minimization) for each particle and compare with the particle's previous best values: If the current value is less than the previous best value, then set the best value to the current value. That is, if $f(X_i) < f(P_i) then X_i = P_i$.
4. Determine the current global position minimum among the particle's best positions. That is:$g = \arg\min_{1\leq i\leq M}(f(P_i))$ M is the population size).
5. Compare the current global position to the previous global: if the current global position is less than the previous global position; then set the global position to the current global.
6. For each dimension of the particle, get a stochastic point between P_{id} and P_{gd} :Eq(4)
7. Attain the new position by stochastic equation: Eq(5)
8. Repeat steps (2)-(7) until a stop criterion is satisfied OR a pre-specified numberof iterations are completed.

4- 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 [3].For a given gray level image of size $N \times N$, let the range pool R be the set of the $(N/L)^2$non-overlapping blocks of size $L \times L$ which is the size of encoding unit. Let the counteractivity of the fractal coding be a fixedquantity of 2. Thus, the domain pool makes up the set of $(N-2L+1)^2$overlapping blocks of size $(2L \times 2L)$. For the case of 256×256 image with 8×8 coding size, the range pool contains 1024 blocks of size $8 \times 8[(256/8) \times (256/8) = 1024]$, and the domain pool contains 58081 blocks of size 16×16 $[(256-16+1) \times (256-16+1) = 58081]$.

For each range block r in R, one searches in the domainpool 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 r. At each search entry, the domain block is first down-sampled to 8×8.In fractal coding, it is also allowed a contrast scaling p and a brightness offset q on the transformed blocks. Thus,the fractal affine transformation Φof $u(x, y)$ in D can be expressed as:

$$\Phi \begin{bmatrix} x \\ y \\ u(x,y) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & p \end{bmatrix} \begin{bmatrix} x \\ y \\ u(x,y) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ q \end{bmatrix} \quad (6)$$

Where the 2×2 sub-matrix $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$is one of the Dihedral transformations and $(t_x, t_y)$is the coordinate of thedomain block in the domain pool. In each search entry, there are eight separate MSE computations required to find the index d such that:

$$d = argmin\{MSE((p_k u_k + q_k), v): \quad k = 0,1,\dots,7\} \quad (7)$$

Where

$$MSE(u,v) = \frac{1}{L^2} \sum_{i,j=0}^{L-1} (u(i,j=0) - v(i,j))^2 \quad (8)$$

Here, $p_k$and $q_k$can be computed directly as

$$p_k = \frac{\left[L^2 \langle u_k, v\rangle - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i,j) \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} v(i,j)\right]}{\left[L^2 \langle u_k, u_k\rangle - \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i,j)\right)^2\right]}$$

$$< u_k, v >= \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i,j).v(i,j)$$

$$< u_k, u_k >= \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i,j)^2 \quad (9)$$

$$q_k = \frac{1}{L^2} \left[\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} v(i,j) - p_k \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i,j)\right] \quad (10)$$

As u runs over all of the 58081 domain blocks in D tofind the best match, the termst_x and t_y in (6) can be obtained. Together with d and the specific p and q corresponding this d, the affine transformation (6) is found in 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. To decode, one first makes up the 1024 affine transformations from the compression codes and chooses any image as the initial one. Then, one performs the 1024 affine transforms on the image to obtain a new image and proceeds recursively. According to Partitioned Iteration Function Theorem (PIFS), the sequence of images will converge. The stopping criterion of the recursion is designed according to user's application. The final image is the retrieved image of fractal coding.

5 - Fractal Image Compression using Quantum PSO

Fractal Image Compression is used to search the near-best matches soas to speed up the encoder. Since the evaluated value of the sub-optimum is close to that of the best match, the quality of the retrieved image can be preserved. As discussed in Section 4, the parameters $t_x, t_y, d, p,$ and q constitute the fractal code. In the proposed method, we encode the particle as (tx, ty), which is the position of the domain block. The quantities p and q can be calculated from (9) and (10), and k is searched separately. At each search entry best index,d in (7) can be obtained. Thefitness value of a particle is defined as the minus of the minimal MSE produced$-MSE((p_d u_d + q_d), v)$When the stopping criterion is satisfied, the final $p_g$with the corresponding d, p, and q is the fractal code of the given range block v. The steps of encoding a range block using QPSO are summarized as follows:

Algorithm 1: Fractal Image Compression using Quantum PSO.

Input:$\boldsymbol{v}$ ***range and Pool domain D.***

Output:the contrast scaling p_k and brightnessoffset q_k.

1: Initialize the parameters of QPSO

2: Initialize an array of particles with therandom position.

3: Calculate the $mbest$ by Eq(3).

4: For each particle (tx, ty), update the particle by Eq (4) and Eq (5).

5: Get the domain block at(tx, ty)in the image. Sub-sample the block and denote it by$u$.

6: Calculatep_k and q_k.

7: Find the fitness of the particle corresponding to the best parameter $d$by Eq (7).

8: If the current particle< previous best particle then set the best value to the current value.

9: Determine the current global position minimum among the particle's best positions.

10:Repeat steps(3-9) until a stop criterion is satisfied OR a pre-specified number of iterations are completed

11: Computep_k and q_k by Eq (9) and Eq (10).

12: Return p_k and q_k

6- Experimental Results

This section experiments the proposed algorithm and compares the proposed algorithm with the performance of PSO in fractal image compression. The proposed algorithm is examined on images Lena, Pepper and Baboon with the size of 256×256 and grayscale. 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 results, the PSNR test is performed:

$$\mathrm{PSNR} = 10 \times \log\left(\frac{255^2}{\frac{1}{m \times n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(f(i,j) - g(i,j)\right)^2}\right) \quad (11)$$

Where $m \times n$ is the size of image. In our experiments, the population size of the swarm is set to be 10 and the maximum number of iterations is set to be 10. Table 1 shows the experimental results on the proposed method, Full Search method and PSO method. According to Tables 1, the proposed algorithm improves the performance of fractal image compression for all the experimental results.

Table 1. Comparison of proposed method with full search and PSO method

Image	Method	PSNR	Time
Lena	Full Search	28.91	3135
	PSO	28.65	64
	QPSO	26.41	59
Pepper	Full Search	29.84	3145
	PSO	28.35	64
	QPSO	26.31	58
Baboon	Full Search	20.15	2966
	PSO	19.77	64
	QPSO	18.02	59

7- Conclusion

In this paper, a quantum particle swarmoptimization (QPSO) in the fractal image compression has been successfully introduced to minimize time. The premature convergence of the traditional PSO algorithm is weakened and the convergence of it is also accelerated. Furthermore, the performance to accomplish the global optimization is also improved subject to the chaos method. Such a method can speed up the encoder and also preserve the image quality. The simulation shows that the encoding time of our method faster than that of the full search method and traditional PSO, whereas the retrieved Lena image quality is still relatively acceptable.

8- References

[1]M. F. Barnsley, J. H. Elton, and D. P. Hardin, "Recurrent iterated function systems," Constructive approximation, vol. 5, pp. 3-31, 1989.

[2]A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," IEEE transactions on image processing, vol. 1, pp. 18-30, 1992.

[3]Y. Fisher, "Fractal image compression," Fractals, vol. 2, pp. 347-361, 1994.

[4]R. Boss and E. Jacobs, "Archetype classification in an iterated transformation image compression algorithm," in Fractal image compression, ed: Springer, 1995, pp. 79-90.

[5]S. Lepsøy and G. E. Øien, "Fast attractor image encoding by adaptive codebook clustering," in Fractal image compression, ed: Springer, 1995, pp. 177-197.

[6]G. Øien, S. Lepsøy, and T. Ramstad, "Reducing the complexity of a fractal-based image coder," in Proc. Eur. Signal Proc. Conf.(EUSIPCO), 1992, pp. 1353-1356.

[7]D. M. Monro and F. Dudbridge, "Fractal approximation of image blocks," in Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on, 1992, pp. 485-488.

[8]J. Li and R. M. Gray, "Context-based multiscale classification of document images using wavelet coefficient distributions," IEEE Transactions on Image Processing, vol. 9, pp. 1604-1616, 2000.

[9]H. J. Yu and J. B. Ra, "Fast triangular mesh approximation of surface data using wavelet coefficients," The Visual Computer, vol. 15, pp. 9-20, 1999.

[10]J.-H. Zhai, X.-Z. Wang, and S.-F. Zhang, "Rough-neural image classification using wavelet transform," in Machine Learning and Cybernetics, 2007 International Conference on, 2007, pp. 3045-3050.

[11]R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, 1995, pp. 39-43.

[12]Y. Shi and R. C. Eberhart, "Particle swarm optimization: developments, applications and resources," in Proceedings of the 2001 congress on evolutionary computation, 2001, pp. 81-86.

[13]J. Kennedy, "Particle swarm optimization," in Encyclopedia of machine learning, ed: Springer, 2011, pp. 760-766.

[14]R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," Swarm intelligence, vol. 1, pp. 33-57, 2007.

[15]M. Clerc, Particle swarm optimization vol. 93: John Wiley & Sons, 2010.

[16]J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in Evolutionary Computation, 2004. CEC2004. Congress on, 2004, pp. 325-331.

[17]M.-S. Wu, J.-H. Jeng, and J.-G. Hsieh, "Schema genetic algorithm for fractal image compression," Engineering Applications of Artificial Intelligence, vol. 20, pp. 531-538, 2007.