International Journal of Academic Research in Computer Engineering
Print ISSN: 2476-7638 and Online ISSN: 2538-2411
Vol. 2, No. 1, Pages. 05-16, May 2018

International Journal of Academic Research in Computer Engineering
Volume 2, Number 1, Pages 05-16, May 2018
www.ijarce.org
IJARCE

Vector Quantization Using the Hybrid Swarm Intelligence Algorithms

Sorayya Amouei[1*], Kamal Mirzaie[2], Fatemeh Saadatjoo[3] and Ali Mohammad Latif[4]

1. Department of Computer Engineering, Science and Art University, Yazd, Iran.
2. Department of Computer Engineering, Maybod Branch, Islamic Azad University, Maybod, Iran.
3. Department of Computer Engineering, Science and Art University, Yazd, Iran.
4. Department of Computer Engineering, Yazd University, Yazd, Iran.

Receive Date 2017.01.21; Accepted Date: 2017.09.01, Published Date: 2018.05.15

*Corresponding Author: S.Amouei (sorayya.amouei@gmail.com)

Abstract
Vector Quantization (VQ) was the powerful technique in image compression. Generating a good codebook is an important part of VQ. There are various algorithms in order to generate an optimal codebook. Recently, Swarm Intelligence (SI) algorithms were adapted to obtain the near-global optimal codebook of VQ. In this paper, we proposed a new method based on a hybrid particle swarm optimization (PSO) and firefly algorithm (FA) to construct the codebook of VQ. The proposed method used PSO algorithm as the initial of FA to develop the VQ. This method is called PSO-FA model. Experimental results indicate that the proposed model is faster than FA. Furthermore, the reconstructed images get higher quality than FA, but it is no significant superiority to the PSO algorithm.

Keywords: Image Compression, Vector Quantization, Swarm Intelligence, Firefly Algorithm, Particle Swarm Optimization Algorithm.

1. Introduction

Nowadays, with the development of social networks and the Internet, the need of users to store and transfer data has increased. Uncompressed information requires high bandwidth; thus, compressing the data became the focus of the interest of many researchers[1, 2]. VQ is a widely used method in compressing digital images. In general, VQ is divided into three phases of codebook generation, vector encoding, and vector decoding. Codebook generation is an important process in diagnosing the efficiency of VQ [3, 4]. In the production phase of codebook, the objective is to find optimum vector-codes to be allocated to training vectors. The operation is conducted by minimizing the gap between the training vectors and their relevant vector- codes. In the coding phase, initially, the image is turned into input vectors (blocks). Then, the input vectors are compared with the obtained code words and the nearest code word is found for each input vector. In other words, the index of nearest code word for each input vector is determined and stored in the index table. In the decoding process, sub-image is restored by the used code word in decoding phase. Finally, after the decoding function, the compressed image is achieved [5, 6].

The production of codebook in VQ method is implemented through various algorithms. Recently, several SI algorithms are used in order to design an optimal codebook in the VQ method[7]. Firefly algorithm (FA) and particle swarm optimization algorithm (PSO) are examples of such these algorithms. With the selection of a suitable codebook, they minimize the amount of distortion error[8].

In this paper, two new hybrid methods are offered and compared namely; FA based on particle swarm optimization and particle swarm optimization algorithm based on firefly algorithm. The purpose of this article is to find a way for better compression rate, savings in storage and faster transfer of information in different networks.

The rest of the paper is organized as follows: In Section 2, related works are discussed. In Section 3, the way of producing codebooks through FA and PSO algorithms is formulated. Section 4, contains an introduction on the way of implementing the proposed methods. In section 4, with respect to PSNR and fitting values, the results of the proposed model are compared with PSO and FA algorithms. In Section 5, Simulation results and comparison with the PSO and FA algorithms are illustrated and finally, conclusions are drawn in Section 6.

2. Related Work

There are several methods to improve offered VQ method. Genetic Algorithm (GA) is one of these methods. In [9] after obtaining the initial population and the selection of two parents, the nearest neighbor algorithm is used for hybrid operations. After combining the operations, mutation operations are applied in order to maintain genetic diversity. Finally, GLA algorithm is used for local optimization.

In [3] using GA, simulated annealing and VQ are offered to encode fractal images. This method was used to estimate the best block range in accordance with the block ranges. The advantage of this method is its high performance in quality and bit rate.

Feng and colleagues in 2007 presented a method for vector quantization based on fuzzy system and PSO algorithm. In [10] Gaussian membership function was selected to calculate the correspondence between the code vectors and image patterns. Therefore, membership function, estimates appropriate weights of the given image pattern by achieving the greatest similarity between vector code and the original pattern of the image. One of the challenges of fuzzy inference system is the reliance of the combination of the best parameters' setting on the forms and the range of the original training image. In order to reduce the examinations and fuzzy inference system's errors, PSO can be used to achieve image compression. Being based on adaptive learning scheme, PSO, and flexible membership function, Fuzzy Inference System can be regarded as a compression image system based on FPSOVQ.

In line with [10] Kapoor and colleagues introduced a fuzzy self-comparative particle swarm optimization learning algorithm to achieve optimal codebook for quantization[11]. This method has combined the advantages of the fuzzy inference method, the concept of basic vector quantization and efficiency of self-adaptive PSO to achieve an optima codebook. The proposed scheme with the help of self-comparative strategy improves the convergence speed and avoids the bonding of global optimum in the local optimum.

In 2014, based on the combination of two algorithms of PSO and K-means, an approach was introduced to solve the problem of local optimum of K-means algorithm. In a way that the application of the result of PSO algorithm as a cluster or initial population of K-means, an optimal method with less distortion error can be achieved[8].

Ming in [12] offered a method of vector quantization based on firefly algorithm. In this method fireflies or solutions are displayed as codebooks. The key point in producing a solution or optimal codebook is to find a codebook with maximum fitness function for all input vectors.

In the FA algorithm, fainter firefly moves toward the brighter firefly and when they have the same brightness, this movement is done randomly. Random motion may reduce the brightness of a firefly, so in [13] an improved method was presented to make Firefly move toward the more intense brightness instead of their random motion.

3. Swarm Intelligence Algorithms

In VQ method, swarm intelligence algorithm is used to generate optimal codebook. Among these algorithms, PSO and FA have been focused more than others. These algorithms are implemented according to the collective behaviors of birds, fish, and fireflies. Although they operate independently, coordinated collective behavior of them leads to the achievement of the desired results[12]. PSO and basic FA algorithms are used in the proposed method. So they are introduced in this section.

3.1. PSO Algorithm

Kennedy and Eberhart offered an algorithm called PSO algorithm. The algorithm is inspired by the group search for food by birds or fish[14]. In this method, each particle represents a solution to the problem and the optimal solution is produced using a particle population. PSO features include ease of implementation and higher speed compared to other evolutionary methods such as genetic algorithms. In the PSO algorithm, each particle i with a mass of m in the search space of D dimensions has a position of $x_i(t)$ and speed of $v_i(t)$. The best global position is calculated by Gbest and the best personal position is calculated by pbest at any moment in time. Direction of each particle is modeled by gbest and pbest [15]. Equation (1) indicates the particle's velocity.

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 \left(p_{id}(t) - x_{id}(t)\right) + c_2 r_2 \left(g_{id}(t) - x_{id}(t)\right) \quad (1)$$

where ω is the coefficient of the particle's inertia; c_1 and c_2 are acceleration coefficients with the common value of 2. To randomize the nature of speed, c_1 and c_2 coefficients are multiplied by the random numbers of r_1 and r_2. After determining the velocity of the particles, the particles' position is updated according to equation (2).

$$x(t) = x(t-1) + v(t) \quad (2)$$

Steps of the standard PSO algorithm are shown in Figure 1.

1. For each particle
2. Initialize particle
3. Do
4. For each particle
5. Calculate fitness value
6. If the fitness value is better than the best fitness value (pBest) in history set current value as the new pBest
7. End
8. Choose the particle with the best fitness value of all the particles as the gBest
9. For each particle
10. Calculate particle velocity
11. Update particle position
12. End
13. While maximum iterations or minimum error criteria is not attained

Figure 1. Pseudo code of the PSO algorithm.

3.2. FA algorithm

Yang introduced the algorithm of fireflies in 2009. Fireflies produce short rhythmic light with different light patterns; they use the lights for the attraction of mates, hunting, and protection. Rhythmic lights, the rate of reflection of sunlight, and the temporal distance between the light signals attract mates to each other. Light intensity 'I' decreases with an increase in the distance 'r' being shown with $I \propto 1/r^2$. Light can be formulated as an optimized objective function. Generally, firefly algorithm has three general assumptions:

- All fireflies are single-sex being attracted to each other regardless of gender.
- Attractiveness is relative to their brightness. In other words, fainter firefly is attracted to the brighter one. If a firefly is not brighter than the other one, it will move randomly.
- The brightness of fireflies is determined by the value of the objective function.

The attraction of a firefly strongly depends on the light seen by nearby fireflies. Equation (3) calculates the absorption of fireflies.

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (3)$$

In a way that β_0 is the absorption in $r = 0$ and γ is the Light absorption coefficient in the air. The distance between two fireflies of i and j at the point x_i and x_j is obtained according to equation (4) as the Euclidean distance.

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k}\right)^2} \quad (4)$$

Such that $x_{i,k}$ is the kth section of spatial coordinate of firefly i or x_i. The attraction of firefly i to brighter firefly j is obtained through equation (5).

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}\left(x_i - x_j\right) + \alpha\left(rand - 1/2\right) \quad (5)$$

In the relation (5) the first statement represents the current position of the firefly i, the second statement and represents the attraction process, and the third statement is the randomization term being performed by α randomization parameter. $rand$ is a random number between [0,1]. β_0 indicates the attractiveness rate in the light source. In most of the relations, these conditions are supposed $\beta_0 = 1$ and $\alpha \epsilon [0,1]$. γ parameter is determined according to the changes in the attraction. This parameter is very effective in the determination of the speed of convergence and the behavior of the firefly algorithm. In theory $\gamma \epsilon [0, \infty)$; thus, in most applications a value is considered between 0.01 to 100.

The procedure for implementing the FA can be summarized as the pseudo code revealed in Figure 2.

1. Begin
2. Objective function $f(x)$, $x = (x1, \ldots, xd)^T$
3. Generate initial population of n fireflies xi, i = 1,2,..,n
4. Formulate light intensity I so that it is associated with $f(x)$
5. While (t <MaxGeneration)
6. Define absorption coefficient γ
7. for i = 1:n (n fireflies)

```
        for j = 1:n (n fireflies)
            if (Ij > Ii),
                move firefly i towards j
            end if
            Vary attractiveness with distance r via
            exp(−γr^2)
            Evaluate new solutions and update light intensity
        end for j
    end for i
    Rank the fireflies and find the current best
    end while
    Post-processing the results and visualization
end
```

Figure 2. Pseudo code of the firefly algorithm

4. The Proposed Hybrid Theory

In the proposed method, VQ codebook is generated by FA-PSO and PSO-FA algorithms. As mentioned in Section III, initial population is randomly generated in FA and PSO algorithms. One of the weaknesses of FA algorithm is being trapped in the local optimum which results in a lower quality reconstruction of the image. In this paper, to overcome this weakness, the PSO algorithm is combined with the FA algorithm. In other words, the result of the implementation of the implementation of the PSO algorithm is considered as the initial population of FA algorithm. The aim is to present an improved FA algorithm because the PSO algorithm, having local optimized parameters, global optimization, and stronger speed of FA acts more powerful than FA and produces better codebooks. In addition, to evaluate the effect of FA algorithm through PSO, the method of FA-PSO is provided, where the result of the implementation of FA algorithm is used as the initial population of PSO algorithm. At first, the operation of the original image's compression has to be turned into the training vectors. For this purpose, the original image is divided into pixels with the size of h × w and blocks with the size of n × n. Suppose L is equal to n × n; so the original image is divided into training vectors including Nb blocks. Each block comprises a set of input vectors. Implementation process of vector quantization being based on the proposed algorithm is shown in Figure 3.

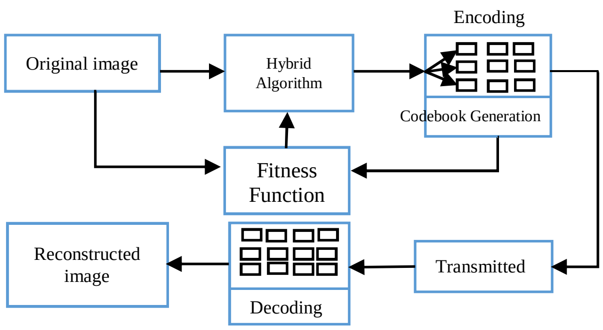


Figure 3. Flow chart of proposed algorithm for VQ

According to Figure 1, the original image being transformed by the training vectors is given to the hybrid algorithm as an input. The proposed algorithm is to find the answer to the suitable fitness function. The answer is the optimal codebook. Decoding is conducted with the help of the produced codebook and the original image. In a way that a code word is assigned to each training vector. In the decoding stage of the image, coding operations is carried out and the reconstructed image is obtained by applying the codebook to training vectors. Figure 4 is an overview of the population structure and solutions in hybrid algorithm. $X_{i,j}$ represents the number j code word in X_i codebook. The first solution in the initial population is defined as X_1 and the last solution is defined as X_S.
Three different scenarios are provided in the following to implement the hybrid algorithm. In these scenarios, the focus has been on the way of performing iterating operations on hybrid algorithms. In general, in all scenarios, the total number of iterates are equal but differ on the way of implementation of replications.

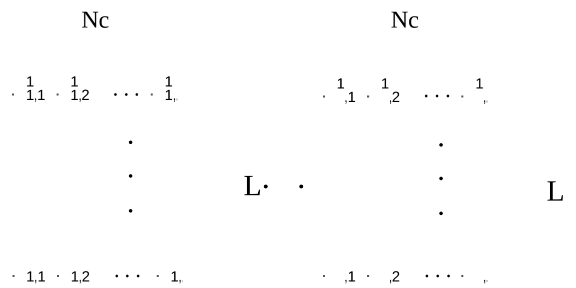


Figure 4. The structures of solutions used in PSO and FA algorithm.

4.1. The First Scenario

In this model, the PSO and the FA algorithms are so combined that initially the maximum cycle number and current cycle number (L) are determined. Next, the first algorithm randomly creates the initial population and is implemented once. Then its output as the input or initial population is given to the second algorithm. Then, after performing the implementation of the second algorithm for once, the termination condition of the algorithm is reviewed. The algorithm is terminated if the current cycle number (L) reaches to the maximum cycle number (MCL). Otherwise, a unit is added to L and the resulting solutions from the second algorithm are given to the first algorithm as input and a recursive loop is generated. The flowchart of the first scenario is given in Figure 5 below.

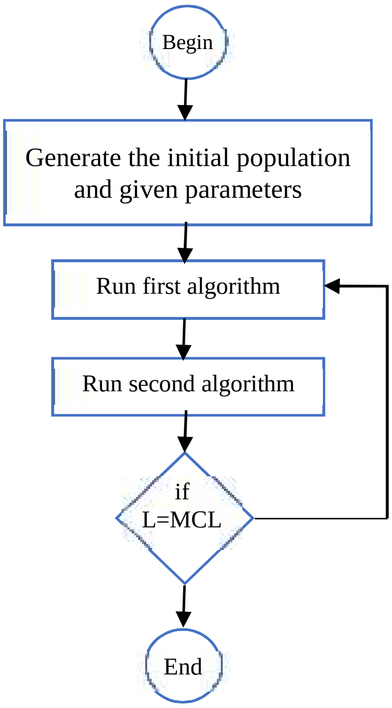


Figure 5. Flow chart of hybrid algorithm for the first scenario

4.2. The Second Scenario

The pattern generates a recursive loop for each PSO and FA algorithms. It should be noted that the number of cycles of the algorithms is the same. The general trend is that initially the total number of cycles of FA and PSO algorithms is determined. Next, the algorithm randomly generated initial population and the prescribed number of repetitions performed. In the next stage, the initial population of the first algorithm is generated randomly and implemented as the number of cycles. The obtained solutions are given to the second algorithm as an input or initial population. The second algorithm is also determined according to the number of performed cycles and the obtained answer the regarded as the final answer. Figure 6 shows the flowchart of the second scenario.

4.3. The Third Scenario

This scenario is similar to the second scenario with the exception that the numbers of cycles are different in each algorithm. In other words, factor 0.7 is considered for the number of the cycles of the PSO algorithm and factor 0.3 is considered for the number of the cycles of the FA algorithm. Flowchart of the third scenario being similar to the second scenario is provided in Figure 6.

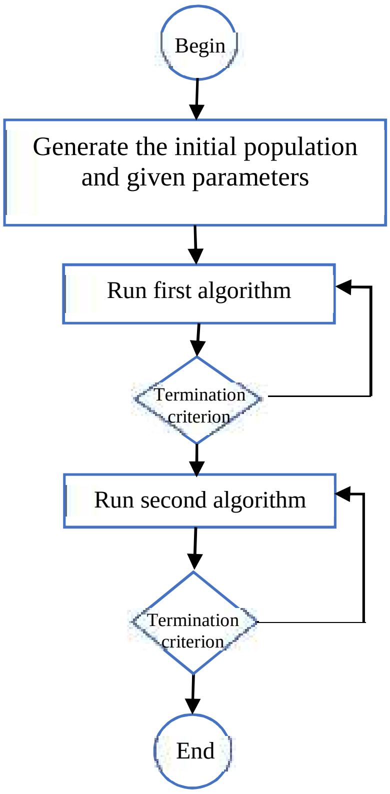
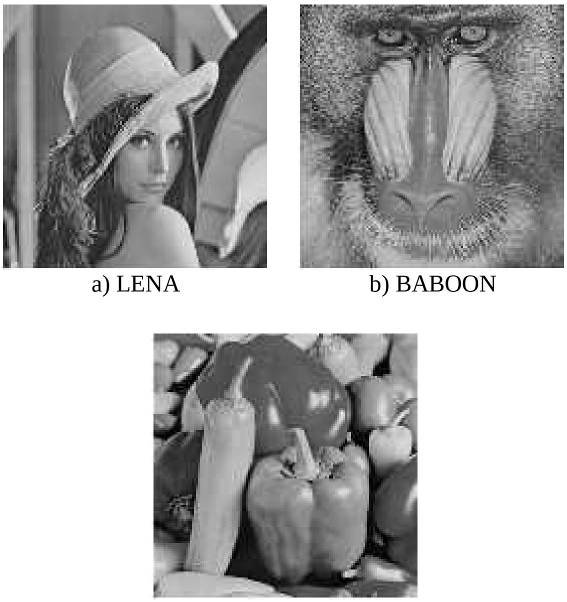


Figure 6. Flowchart of hybrid algorithm for the second and third scenario

5. The Results of Implementation

The compression operation is conducted on three gray images named “LENA”, “BABOON” and “PEPPERS” are shown in Figure 7. The size of each image is 512 × 512 pixels and their amplitude resolution is 8 –bits.



a) LENA b) BABOON

c) PEPPERS

Figure 7. The Test Images: (a) LENA, (b) BABOON and (c) PEPPER

According to Figure 8 the original image is first divided into 4 × 4 pixels non-overlapping blocks and each block is turned into a 16 dimensional vector. The input image is displayed through 16384 training vectors. Six codebooks in 8, 16, 32, 64, 128 and 256 sizes were implemented in the experiments. Table 1 and Table 2 show the parameters of the FA and PSO algorithms.

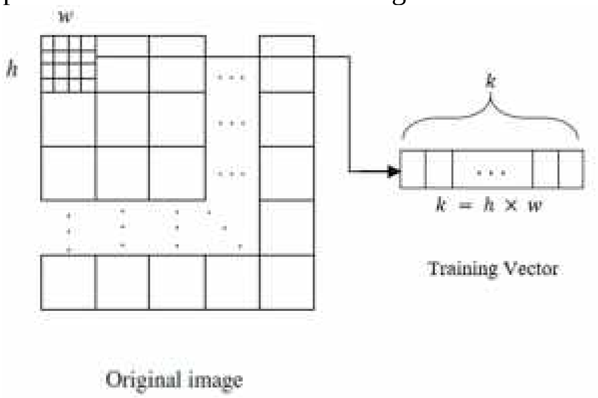


Figure 8. Converting original image into training vectors

Table 1. The parameters used in the FA algorithm

parameter	explanation	value
S	Number of particles	100
ITER	Number of iterations	50
α	Mutation coefficient	0.01
β0	Initial attractiveness coefficient	1.0
γ	light absorption coefficient	1.0

Table 2. The parameters used in the PSO algorithm

parameter	explanation	value
S	Number of particles	100
V	Velocities randomly	[-1,1]
Iter	Number of iterations	50
ϕ1	Cognitive coefficient	2.1
ϕ2	Cognitive coefficient	2.0
ϕ	ϕ1+ϕ2	4.1
K	$K = \frac{2}{\lvert 2\phi-\sqrt{\phi^2-4\phi}\rvert}$	0.729
C1	ϕ1*K	1.5309
C2	ϕ2*K	1.458

Algorithms achieved in in each scenario are shown with the respective index. For example, the algorithm derived from the combination of PSO-FA in the first scenario is called 1-PSO-FA. These algorithms are compared with each other; PSO, FA, 1-PSO-FA, 1-FA-PSO, 2-PSO-FA, 2-FA-PSO, 3-PSO-FA and 3-FA-PSO. Programming of algorithms is based on MATLAB 2013 being designed on PC with 2.5 GHz CPU and 4G RAM. The number of initial solutions is 100 and the number of cycles is 50.

Equation (6) indicates the bit rate at which Nc is the size of the designed codebook and L is the number of blocks' pixels.

$$b\_r = \frac{\log_2 N}{L} \quad (6)$$

To check the quality of the encoded image, the equation (7) is used, called PSNR.

$$P = 10\log_{1}\left(\frac{(h-1)(w-1)}{M}\right) \quad (7)$$

MSE is the mean square error between the original image and the reconstructed image being defined by equation (8).

$$M = \frac{1}{h w}\sum_{i=1}^{h}\sum_{j=1}^{w}\left(y_{i} - \overline{y_{ij}}\right)^2 \quad (8)$$

h×w is the image size, y_{i} and $\overline{y_{i}}$ are Pixel values at the position of (i,j) in the original and reconstructed image. Eight algorithms were implemented in experiments and each for ten times. The mean of the ten-times implementation is considered as the main answer.

In experiments, the algorithms are implemented and then executed 10 times. Figure 9 to Figure 11 show the average PSNR value of each of the test images. These results reveal that the average PSNR of FA algorithm is the worst and the other four algorithms can effectively improve the results of FA algorithm. By examining the three training images it is concluded that all the hybrid PSO-FA algorithms in three scenarios, have a higher PSNR than FA algorithm. This result suggests that the weakness of the FA algorithm can be improved by selecting the appropriate initial population. Using the result of the powerful implementation of the PSO algorithms as the initial population of FA improves the FA algorithm.

In contrast, according to Figure 12 to Figure 14 hybrid algorithms of FA-PSO in three scenarios have a PSNR which is similar to PSO; this leads to the improvement of this algorithm. In other words, the initial population generated by the FA algorithm has had no effect on the performance of the PSO algorithm. This result shows the strength of PSO algorithm compared to the FA.

Examining the different scenarios, in the following, it is realized that the way of performance of the cycles in PSO and FA algorithm has little impact on the performance of FA FA-PSO and PSO-FA hybrid algorithms. FA-PSO algorithms in three scenarios indicate a fairly similar PSNR. This is also true for PSO-FA hybrid algorithms.

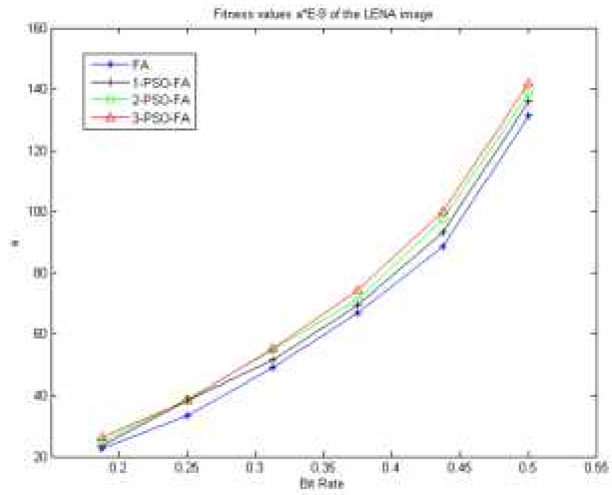


Figure 9. The Average of Peak Signal-to-Noise Ratio (PSNR) of LENA Image of the four Different Vector Quantization Methods

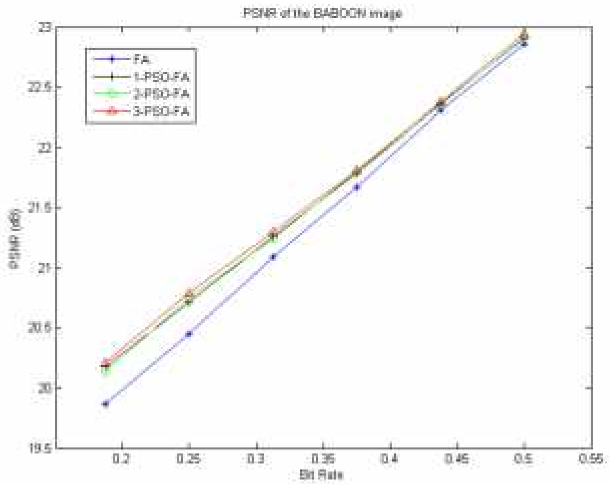


Figure 10. The Average of Peak Signal-to-Noise Ratio (PSNR) of BABOON Image of the four Different Vector Quantization Methods

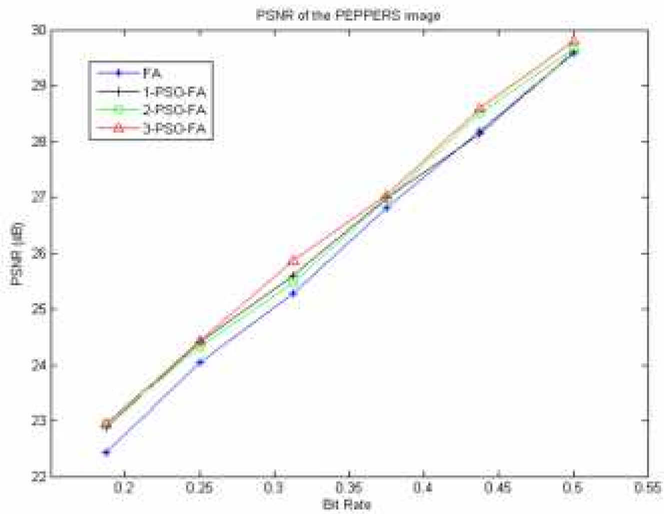


Figure 11. The Average of peak Signal-to-Noise Ratio (PSNR) of PEPPERS Image of the four Different Vector Quantization Methods

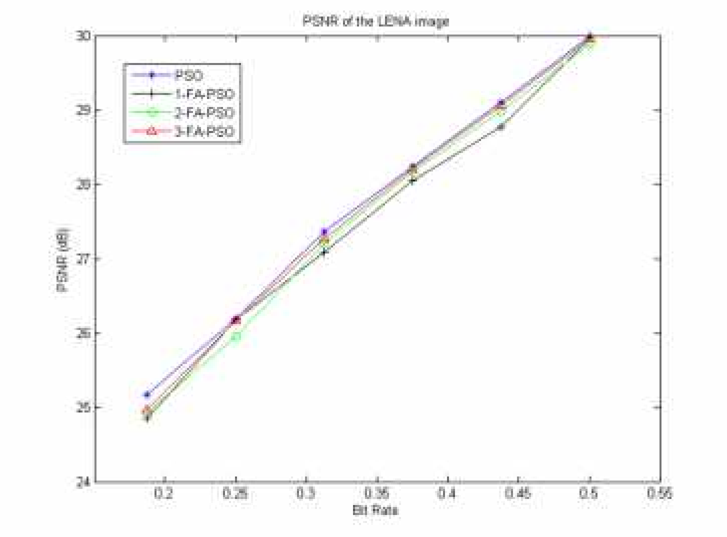


Figure 12. The Average of Peak Signal-to-Noise Ratio (PSNR) of LENA Image of the four Different Vector Quantization Methods

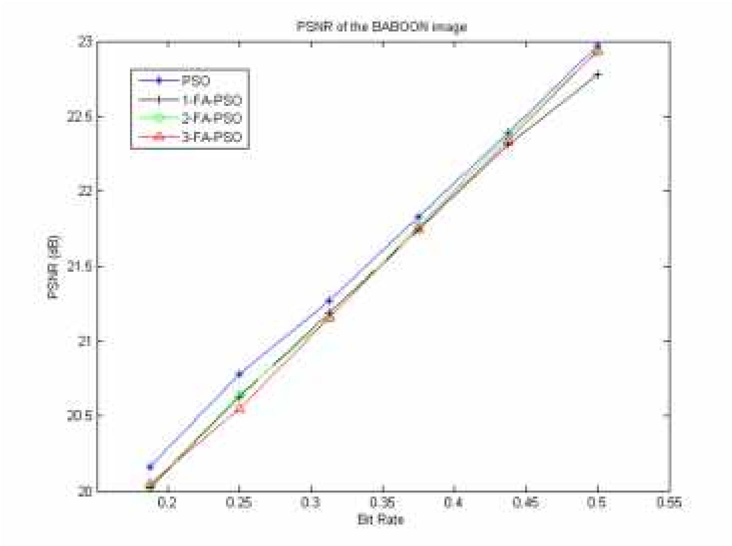


Figure 13. . The Average of Peak Signal-to-Noise Ratio (PSNR) of BABOON Image of the four Different Vector Quantization Methods

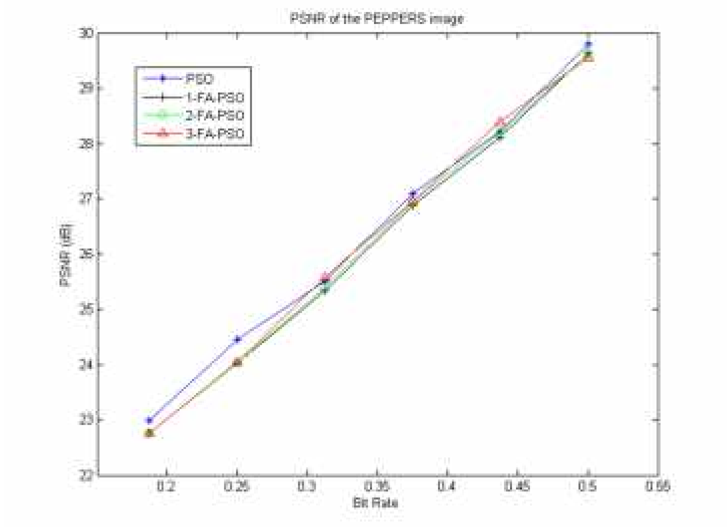


Figure 14. The Average of Peak Signal-to-Noise Ratio (PSNR) of PEPPERS Image of the four Different Vector Quantization Methods

The average fitness values of the three test image by using the four vector quantization algorithms are shown as Figure 15 to Figure 17. These experimental results demonstrated that the fitness of the three test images using the Hybrid algorithms of PSO-FA in three scenarios, have a much better fitness than the basic FA algorithm. However, according to Figure 18 to Figure 20 the hybrid algorithms FA-PSO have almost the same fitness value when compared to the PSO algorithm. The results show that hybrid PSO-FA algorithm improves the FA algorithm; however, FA-PSO algorithm does not improve the PSO algorithm. In the three scenarios, all hybrid FA-PSO algorithm have a similar fitness. The result is the same for hybrid algorithms PSO-FA algorithm in different scenarios.

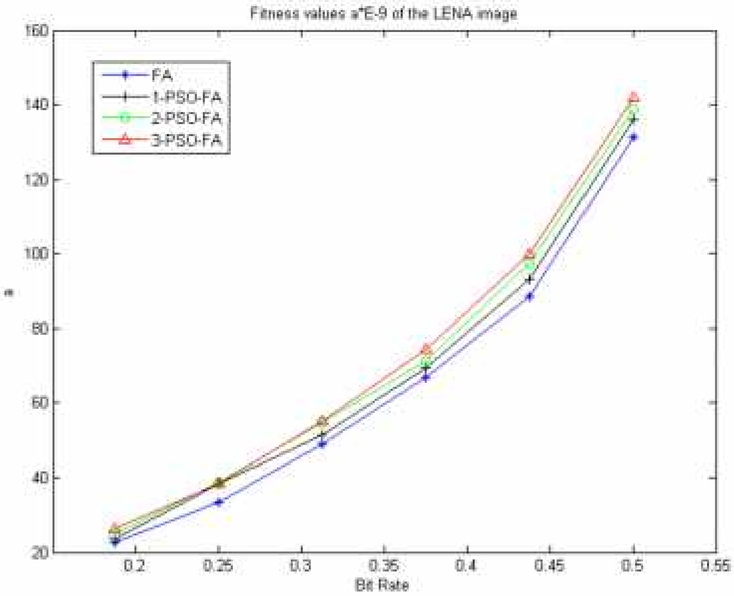


Figure 15. The average of fitness values ($\alpha\times10^9$) of LENA image of the four different vector quantization methods

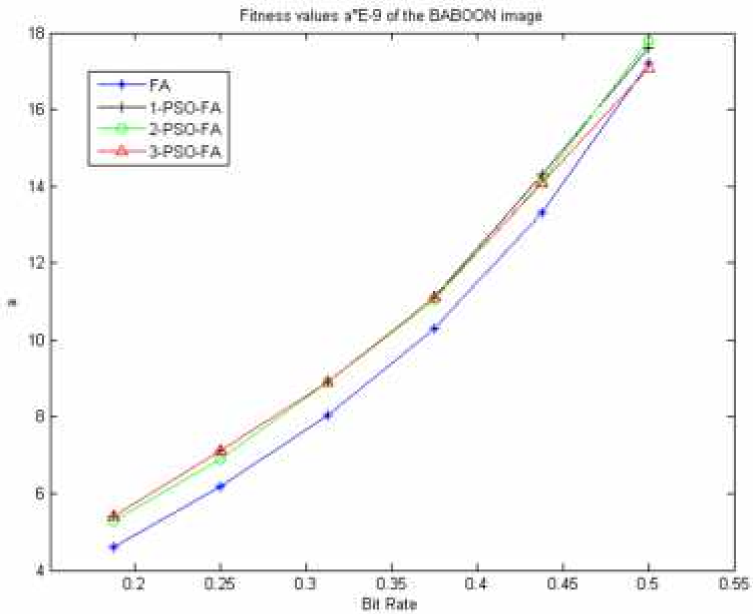


Figure 16. The average of fitness values ($\alpha\times10^9$) of BABOON image of the four different vector quantization methods

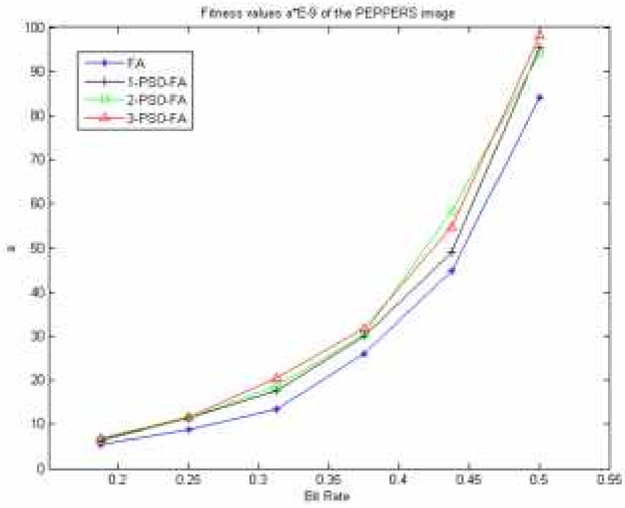


Figure 17. The average of fitness values ($\alpha\times10^9$) of PEPPERS image of the four different vector quantization methods

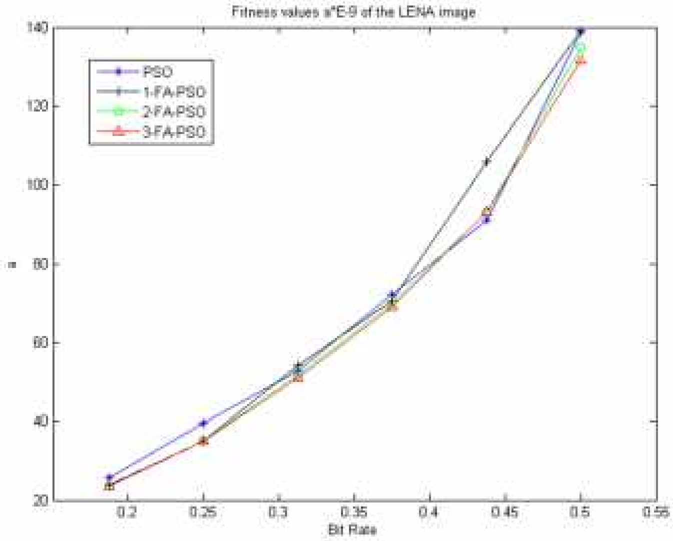


Figure 18. The Average of Fitness Values ($\alpha\times10^9$) of LENA Image of the four Different Vector Quantization Methods

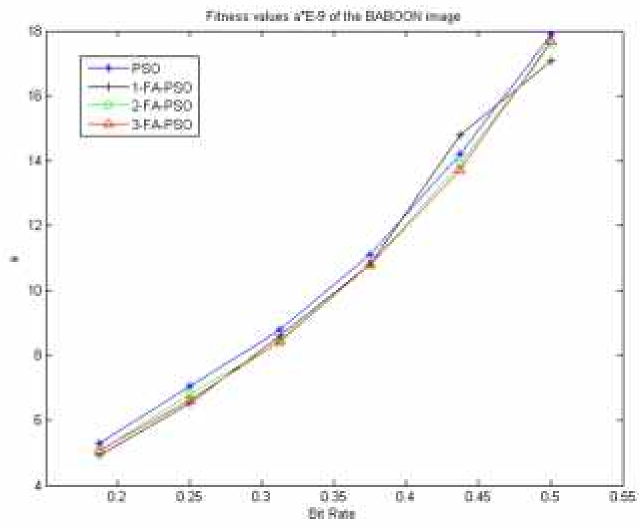


Figure 19. The Average of Fitness Values ($\alpha\times10^9$) of BABOON image of the four Different Vector Quantization Methods

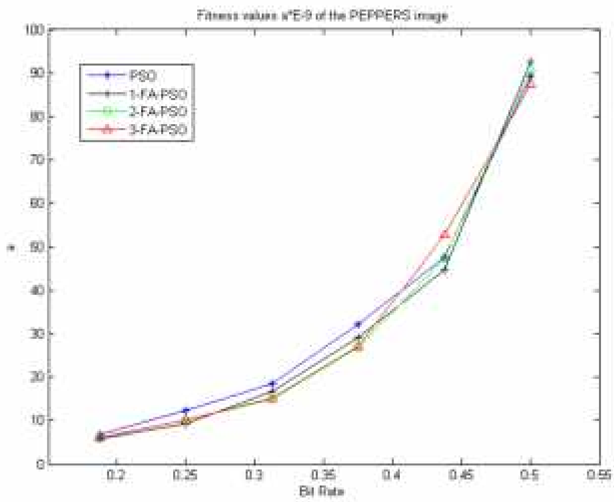


Figure 20. The Average of Fitness Values ($\alpha \times 10^9$) of PEPPERS Image of the four Different Vector Quantization Methods

Figure 21 shows the reconstructed training images by the codebook with size of 256 for 3-FA-PSO algorithm.

Figure 21. Original and Reconstructed Images Made by 3-FA-PSO Algorithm for Codebook with Size of 256

6. Conclusion

In this paper, eight algorithms were studied; PSO, FA, 1-PSO-FA, 1-FA-PSO, 2-PSO-FA, 2-FA-PSO, 3-PSO-FA and a 3-FA-PSO. The results showed that the hybrid algorithm of PSO-FA improves the performance of basic FA algorithm and provides reconstructed image with a better quality. FA-PSO hybrid algorithm based on three scenarios and PSO have a similar behavior when the size of the codebook is increased. Comparing the hybrid FA-PSO algorithms in three scenarios, it is realized that they have the same performance. In other words, the implementation of the cycles in the basic algorithms has had little effect on the final results and only with the same number of cycles; the same result can be achieved. This is true in hybrid PSO-FA algorithms in the three scenarios. By comparing the two hybrid algorithms of PSO-FA and FA-PSO in the three scenarios it is shown that the PSO-FA algorithm has better performance with optimal codebook. The reason for this superiority is the strength of PSO algorithm to FA algorithm. In a way that if the algorithm is started with PSO, it will have better results. This result indicates a high impact of the initial population on the performance of algorithms. It should be noted that because of the same number of the total cycles of all scenarios, the FA-PSO and PSO-FA algorithms have the same implementation times.

References

[1] V. Bhaskaran and K. Konstantinides, Image and video compression standards: algorithms and architectures vol. 408: Springer Science & Business Media, 1997.

[2] R. C. Gonzalez and R. E. Woods, "Image processing," Digital image processing, vol. 2, 2007.

[3] A. N. B. Kamal, "Iteration Free Fractal Image Compression For Color Images Using Vector Quantization, Genetic Algorithm And Simulated Annealing," The Online Journal of Science and Technology-January, vol. 5, 2015.

[4] H. Kekre, P. Natu, and T. Sarode, "Color Image Compression Using Vector

Quantization and Hybrid Wavelet Transform," Procedia Computer Science, vol. 89, pp. 778-784, 2016.

[5] X. Li, J. Ren, C. Zhao, T. Qiao, and S. Marshall, "Novel multivariate vector quantization for effective compression of hyperspectral imagery," Optics Communications, vol. 332, pp. 192-200, 2014.

[6] H. Sun, K.-Y. Lam, S.-L. Chung, W. Dong, M. Gu, and J. Sun, "Efficient vector quantization using genetic algorithm," Neural Computing & Applications, vol. 14, pp. 203-211, 2005.

[7] D. Karaboga and B. Akay, "A survey: algorithms simulating bee swarm intelligence," Artificial Intelligence Review, Vol. 31, pp. 61-85, 2009.

[8] G. Saini and H. Kaur, "A novel approach towards K-mean clustering algorithm with PSO," Int. J. Comput. Sci. Inf. Technol, vol. 5, pp. 5978-5986, 2014.

[9] P. Fränti, "Genetic algorithm with deterministic crossover for vector quantization," Pattern Recognition Letters, Vol. 21, pp. 61-68, 2000.

[10] H.-M. Feng, C.-Y. Chen, and F. Ye, Evolutionary fuzzy particle swarm optimization vector quantization learning scheme in image compression, Expert Systems with Applications, Vol. 32, pp. 213-222, 2007.

[11] M. Kumar, R. Kapoor, and T. Goel, "Vector Quantization based on Self-Adaptive Particle Swarm Optimization, International Journal of Nonlinear Sciences, vol. 9, pp. 311-319, 2010.

[12] M.-H. Horng, Vector quantization using the firefly algorithm for image compression," Expert Systems with Applications, Vol. 39, pp. 1078-1091, 2012.

[13] K. Chiranjeevi, U. R. Jena, B. M. Krishna, and J. Kumar, "Modified Firefly Algorithm (MFA) Based Vector Quantization for Image Compression," in Computational Intelligence in Data Mining, Vol. 2, ed: Springer, 2016, pp. 373-382.

[14] J. a. E. Kennedy, R. C, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

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