ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL METHOD FOR FRACTAL IMAGE COMPRESSION USING POLYNOMIAL HYBRID WAVELET AND PARTICLE SWARM OPTIMIZATION

[1]D.VENKATASEKHAR, [2]P.ARUNA, [3]V.DHIVYA

[1]Associate professor, Department of CSE, Annamalai University
[2]Professor, Department of CSE, Annamalai University
[3]M.E Student, Department of CSE, Annamalai University

E-email: [1]ramavenkatasekar@yahoo.co.in, [2]arunapuvi@yaoo.co.in , [3]dhivyavageesan@gmail.com

ABSTRACT

Fractal compression is a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Fractal Image Compression (FIC) techniques take more time to perform processes are encoding and global search. Particle Swam Optimization (PSO) and Wavelet transformation methods are used to reduce the encoding time. Both of the techniques and their performance are analyzed in terms of their compression ratio, encoding time, Mean Square Error (MSR) and PSNR (Peak Signal-to Noise Ratio) value. Based on these parameters the performances of the techniques are studied and a comparative analysis of these techniques is provided.

Keywords: *Compression Ratio; Discrete Cosine Transform; Discrete Wavelet Transform; Encoding Time; Fractal Image Compression; Particle Swarm Optimization.*

1. INTRODUCTION

Fractal Image Compression (FIC) is one of the widely used image processing application in image retrievals, image signature, texture segmentation, feature extraction [1]. The overhead expense in fractal encoding is due to huge number of Range -domain compression required to find best matching pair. The basic idea of fractal compression is to find similarities between larger and smaller portions of an image. This is accomplished partitioning the original attractor after little iteration [2]. Fractal compression allows fast decompression but has long encoding times. The most time consuming part is the domain blocks searching from each range [3, 4].

Fractal compression based on wavelet transform –here original image is converted to frequency domain image with use of wavelet transform. Form fractal coding file to reduce low frequency coefficient and high frequency coefficient [6-10]. PSO is a general-purpose optimization algorithm which also uses the concept of fitness [11].

In this paper, the proposed method has benefits in reducing encoding time and increasing compression ratio compared with other image compression techniques, such as Discrete Wavelet Transform[5], Discrete Cosine Transform[7], and Particle Swarm Optimization [12,13].

2. DISCRETE WAVELET TRANSFORM

The wavelet transform represents a signal with a good resolution in both time and frequency. Wavelet transform decomposes a signal into a set of basic functions. These basis functions are called *wavelets.*

2.1 Decomposition of a two dimensional mage

The wavelet transform of a two dimensional image is essentially the two

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

dimensional discrete wavelet transform of the image. The definition of the two dimensional wavelet transform is:

$$WT_f(a, b_x, b_y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x, y)\varphi^*_{a,b_x,b_y}(x, y)\,dxdy, \quad (1)$$

Where

$$\varphi_{a,b_x,b_y}(x, y) = \frac{1}{|a|}\varphi\left(\frac{x-b_x}{a}, \frac{y-b_y}{a}\right) \quad (2)$$

Where b_x and b_y are the transformation of the two.

The wavelet coefficients of image after wavelet transform can compose three kinds of oriented wavelet subtrees: the horizontal direction (H) wavelet subtree which has low frequency in horizontal direction and high frequency in vertical direction, the vertical direction (V) wavelet subtree which has high frequency in horizontal direction and low frequency in vertical direction; the diagonal direction (D) wavelet subtree which has both high frequency in horizontal and vertical direction.

The contractive mapping operations carried out in the spatial domain have direct analogy in the wavelet domain. The averaging and sub sampling operation S matches the size of the domain tree with that of the range tree. If we use Haar wavelet transform, the sub sampling operation is equivalent to moving up the domain block tree by one scale in the wavelet domain, since the Haar transform is exactly the same as combined averaging and sub sampling operations.

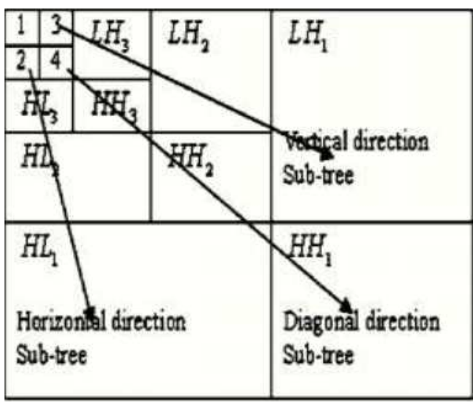

Figure 1: wavelet subtree

Encoding process

1. Take an image as input.

2. Calculate the N-level DWT (Haar).

3. Partition the H, V, D components of the i^{th} level into domain blocks of size 2Bx2B.

4. Partition the H, V, D components of the $(i+1)^{th}$ level into range blocks of size BxB.

5. Find the best matching domain block tree for each range block tree

6. Save the mapping information.

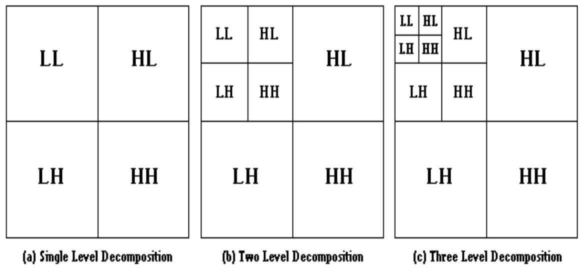

Figure 2: Pyramidal Decomposition of an Image

As mentioned above, the LL band at the highest level can be classified as most important, and the other 'detail' bands can be classified as of lesser importance, with the degree of importance decreasing from the top of the pyramid to the bands at the bottom.

Figure 3: The three level wavelet decomposition of the 'Lena' image.

Multilevel subgraphs are produced by wavelet transform in which different resolutions corresponds to different frequencies. The wavelet decomposition with three level o the Lena image is shown in Figure. 3.

2.2 Quantization

Journal of Theoretical and Applied Information Technology
31st May 2014. Vol. 63 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Quantization converts a sequence of floating numbers w' to a sequence of integer's q. The simplest form is to round to the nearest integer. Another method is to multiply each number in w' by a constant k, and then round to the nearest integer. Quantization is called lossy because it introduces error into the process, since the conversion of w' to q is not one to one function.

3. DISCRETE COSINE TRANSFORM

A discrete cosine transform (DCT) is mainly used various application of compression in image processing. Two dimensional based DCT is mostly used for compression of individual video frames and multidimensional based DCT is used for compression of video stream. The 2-D DCT based compression by decreasing the number of computations, and increasing the accuracy of reconstruction.

The DCT can be extended to the transformation of 2D signals or images. This can be achieved in two steps: by computing the 1D DCT of each of the individual rows of the two dimensional image and then computing the 1D DCT of each column of the image. If represents a 2D image of size *x (n1, n2) N × N*, then the 2D DCT of an image is given by:

$$Y[j,k] = C[j]C[k]\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x[m,n]\cos\left(\frac{(2m+1)j\pi}{2N}\right)\cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad (3)$$

Where $j,k,m,n = 0,1,2,\dots,N-1$ and

$$C[j] \ \ and \ \ C[k] = \begin{bmatrix} \frac{1}{\sqrt{N}} & for\ j,k=0 \\ \frac{1}{\sqrt{N}} & for\ j,k=1,2,\dots,N-1 \end{bmatrix} \quad (4)$$

The DCT presented in equation (4) is orthonormal and perfectly reconstructing provided the coefficients are represented to an infinite precision. This means that when the coefficients are compressed it is possible to obtain a full range of compressions and image qualities. The coefficients of the DCT are always quantized for high compression, but DCT is very resistant to quantisation errors due to the statistics of the coefficients it produces. The coefficients of a DCT are usually linearly quantised by dividing by a predetermined quantisation step.

As shown in the figure 4, the image is first partitioned into non-overlapping 8 × 8 blocks. A Forward Discrete Cosine Transform (FDCT) is applied to each block to convert the spatial domain gray levels of pixels into coefficients in frequency domain. To improve the precision of the DCT the image is 'zero shifted', before the DCT is applied. This converts a 0 → 255 image intensity range to a -128 → 127 range, which works more efficiently with the DCT. One of these transformed values is referred to as the DC coefficient and the other 63 as the AC coefficients.

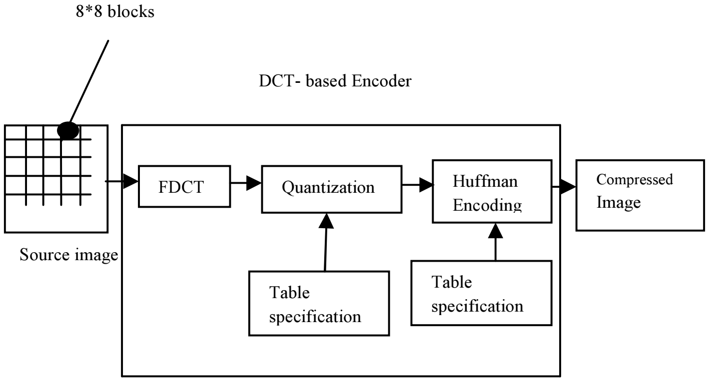

Figure 4: Encoder

3.1 The Process

The following is the general overview of the JPEG process. Later we will go through the detailed tour of JPEG's method so that a more comprehensive understanding of the process may be acquired.

1. The image is broken into 8*8 blocks of pixels.

2. Working from left to right, top to bottom, the DCT is applies to each block.

3. Perform the polynomial representation to the variable blocks sizes that resultant of step2 according to equations (5, 6 and 5).

$$a_0 := \frac{1}{n*m}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} I(i,j) \quad (5)$$

Journal of Theoretical and Applied Information Technology
31st May 2014. Vol. 63 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$a_1 := \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} I(i,j) * (j - x_c)}{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} (j - x_c)^2}$$
(6)

$$a_2 := \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} I(i,j) * (j - y_c)}{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} \left(j - y_c\right)^2}$$
(7)

Where I (i, j) is the original image block of size n×m and

$$x_c := \frac{n-1}{2}$$

$$y_c := \frac{m-1}{2}$$

4. Each block is compressed through quantization.

5. The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.

6. When desired the image is constructed through decompression, a process that uses the Inverse Discrete Cosine Transform (IDCT).

3.2 Quantisation

DCT-based image compression relies on to reduce the data required to represent the image. Quantization is the process of reducing the number of possible values of a quantity, thereby reducing the number of bits needed to represent it.

3.3 Huffman Coding:

Huffman coding is an efficient source coding algorithm for source symbols that are not equally probable. A variable length encoding algorithm was suggested by Huffman in 1952, based on the source symbol probabilities *P* (x_i), *i=1, 2……, L.* The algorithm is optimal in the sense that the average number of bits required to represent the source symbols is a minimum provided the prefix condition is met. The steps of Huffman coding algorithm are given below:

1. Arrange the source symbols in increasing order of heir probabilities.

2. Take the bottom two symbols & tie them together as shown in Figure 5. Add the probabilities of the two symbols & write it on the combined node. Label the two branches with a‘1’ & a ‘0’ as depicted in Figure 5.

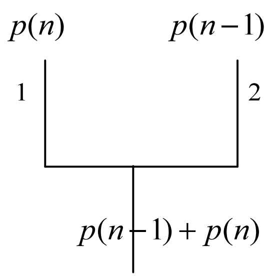

Figure 5: tree

3. Treat this sum of probabilities as a new probability associated with a new symbol. Again pick the two smallest probabilities, tie them together to form a new probability. Each time we perform the combination of two symbols we reduce the total number of symbols by one. Whenever we tie together two probabilities (nodes) we label the two branches with a ‘0’ & a ‘1’.

4. Continue the procedure until only one procedure is left (& it should be one if your addition is correct). This completes the construction of the Huffman Tree.

5. To find out the prefix codeword for any symbol, follow the branches from the final node back to the symbol. While tracing back the route read out the labels on the branches. This is the codeword for the symbol.

4. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a robust stochastic optimization technique based on the movement and intelligence of swarms. It consists of a swarm of particles. Each particle resides at a position in the search space .The

Journal of Theoretical and Applied Information Technology
31st May 2014. Vol. 63 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

fitness of each particle represents the quality of its position [11].The particles flies over the search space with a certain velocity. It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution .Each particle is treated as a point in a N-dimensional space which adjusts its “flying” according to its own flying experience as well as the flying experience of other particles.

In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved. This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained by any particle in the population. This best value is a global best called gbest [12].

Let $\varphi i,j(k)$ and $vi,j(k)$ denote the j^{th} dimensional value of position and velocity of ith particle in the swarm, respectively, at step **k**. The PSO updating rules can be expressed as

$$v_{i,j}(\mathrm{k}) = \mathrm{w}\cdot v_{i,j}(\mathrm{k}-1) + \mathrm{c1}\cdot r1_{i,j}\cdot(\varphi^{*}_{i,j} - \varphi_{i,j}(\mathrm{k}-1)) + \mathrm{c2}\cdot r2_{i,j}\cdot(\varphi^{\#}_{j} - \varphi_{i,j}(\mathrm{k}-1)) + \mathrm{c3}\cdot r3_{i,j}\cdot(\varphi^{\theta}_{j} - \varphi_{i,j}(\mathrm{k}-1)) \quad (8)$$

$$\varphi_{i,j}(\mathrm{k}) = \varphi_{i,j}(\mathrm{k}-1) + v_{i,j}(\mathrm{k}), \quad (9)$$

Where,

W is the inertial weight which controls the effect of velocity at k − 1, φ^{*}_{i} denotes the best position of ith particle up to step **k − 1**, which is named as **pbest**.

The term $\varphi\#$ denotes the best position of the whole swarm up to step **k − 1**, which is named as $gbest$ and,

φΘ Denotes the best in a neighborhood or sub-swarm, named as $nbest$, which is the best particle’s position among all particles in the sub-swarm.

The quantities $r1_{i,j}$, $r2_{i,j}$ and $r3_{i,j}$ are random numbers selected from [0, 1] in jth dimensional position and ith particle of the swarm. The coefficients **c1, c2** and **c3** are positive numbers and they represent the individuality, swarm, and subset of swarm coefficients, respectively.

Steps in PSO:

1. Initialize the swarm from the solution space.
2. Evaluate fitness of each particle.
3. Update individual and global bests.
4. Update velocity and position of each particle using equations (11) and (12).
5. Go to step2, and repeat until termination condition.

Since the fractal encoding scheme searches the best match in the domain pool for every range block, one takes the absolute position of a domain block to constitute a particle. The idea of the neighborhood is the partitioning of the whole swarm [13]. For example, there are 15 particles in the swarm and the size of neighborhood is 5, say. Then the particle 3 would only communicate with particles 1 through 5, and the particle 9 would only communicate with particles 6 through 10. Neighborhood best can avoid falling into local minima in the process of PSO.

The velocity updating formula can be illustrated using student-class-school model. The experience of a student (particle) is not only just influenced by two factors, which are his best experience ($pbest$) and the experience of the global best ($gbest$) student in the school (swarm), but also by the experience of the best student ($nbest$) in the class (sub-swarm).

6. EXPERIMENT AND RESULTS

This section of paper describes the analysis of the results obtained for our research findings. There are many ways to measure the quality of reconstructed image, obtained with a given compression method. Probably the three most popular measures used are Compression Ratio (CR), Peak Signal to Noise Ratio (PSNR)

Journal of Theoretical and Applied Information Technology
31st May 2014. Vol. 63 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and Encoding Time. PSNR is a measure of the peak error. Logically a higher value of PSNR is good because it means that the ratio of signal to Noise is higher. The proposed algorithm has been implemented successfully using Mat lab and evaluated their performance with different images.

a) b)

c) d)

Figure 6:Original Image a)Parrot, b)Rose, c)Car, d)Flower

a) b)

b) d)

Figure 7:Reconstructed Image using DWT a)Parrot, b)Rose, c)Car, d)Flower

a) b)

c) d)

Figure 8:Reconstructed Image using DCT a)Parrot, b)Rose, c)Car, d)Flower

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a) b) c) d)

Figure 9:Reconstructed Image using PSO a)Parrot, b)Rose, c)Car, d)Flower

Table 1: Comparison of DWT and DCT

Image	Method	CR	PSNR	Encoding time(sec)
Parrot	DWT	1.5428	30.8248	4.72073
	DCT	10.4932	30.1586	0.8991
Rose	DWT	2.4380	25.7361	2.1903
	DCT	9.1408	30.2813	0.8652
Car	DWT	3.2223	21.8318	2.4143
	DCT	8.3565	30.4500	1.3151
Flower	DWT	2.4462	29.8134	1.2608
	DCT	11.5785	30.6451	0.9529

Figure 10 show the compression ratio Vs. image compression techniques for different images.

The performance of the DCT method provided higher compression ratio than the DWT method as discussed in the comparative analysis.

Table 2: Comparison of DCT and PSO

Image	Method	CR	PSNR	Encoding time(sec)
Parrot	DCT	10.4932	30.1586	0.8991
	PSO	24.2711	38.3610	0.6762
Rose	DCT	9.1408	30.2813	0.8652
	PSO	16.8864	45.2366	0.2586
Car	DCT	8.3565	30.4500	1.3151
	PSO	12.3554	38.3643	0.2698
Flower	DCT	11.5785	30.6451	0.9529
	PSO	21.6299	50.6044	0.2449

Journal of Theoretical and Applied Information Technology
31st May 2014. Vol. 63 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

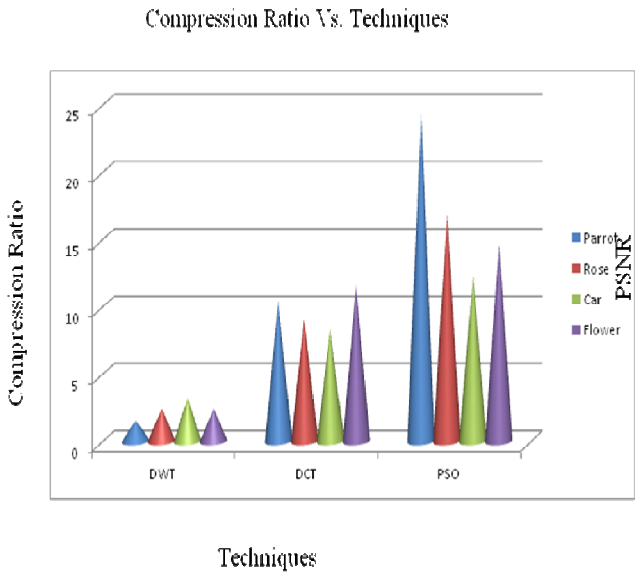

Figure 10: Compression Ratio Versus Techniques

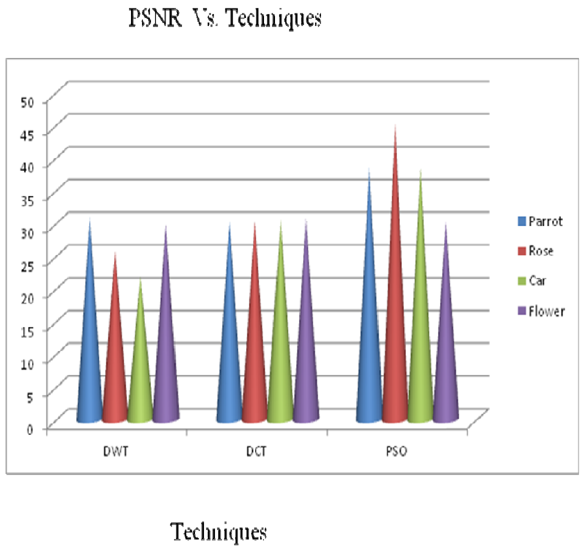

Figure 11: PSNR Versus Techniques

Figure 11 show the Peak Signal –to Noise Ratio Vs. image compression techniques for different images.

Figure 12 show the Encoding Time Vs. image compression techniques for different images.

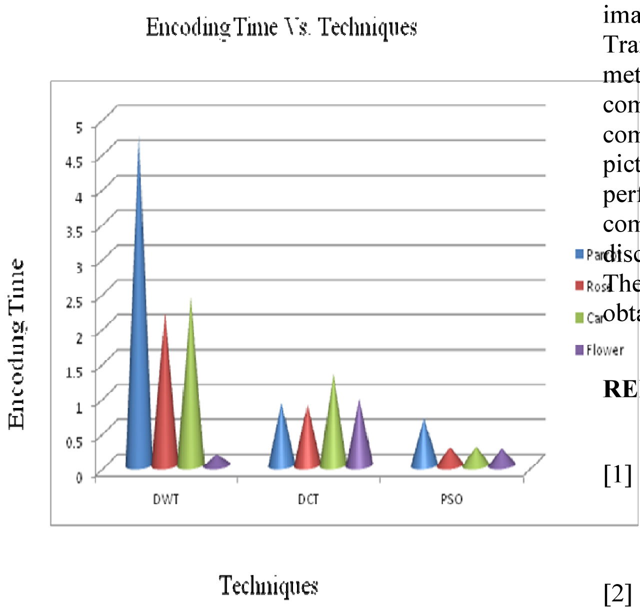

Figure 12: Encoding Time Versus Techniques

7. CONCLUSION

In this paper, the results of digital image was compared which are obtained from fractal image compression using Discrete Wavelet Transform and Discrete Cosine Transform methods. The effect of image contents and compression ratios were examined. This compression provides a better performance on picture quality at higher compression ratio. The performance of the DCT method provided higher compression ratio than the DWT method as discussed in the comparative analysis table 1. Then DCT method compare with PSO method to obtain best one.

REFERENCES

[1] A. E. Jacquin.,"Image Coding based on a fractal theory of iterated contractive image transformation", *IEEE Trans. On Image Processing,* 1(I) :(1992).

[2] Arnaud E. Jacquin, (1993)., "Fractal image coding", *Proceedings of IEEE VOL.81, pp. 1451-1465.*

[3] Hitashi., et.al., "Fractal image compression- a review," *International Journal of Advanced Research in Computer Science*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and Software Engineering, Volume 2, Issue 2, February 2012.

[4] Madaliya, et.al, "Fractal Image Compression Using Graphics Hardware," *International Journal Of Advanced Engineering Research And Studies.*

[5] Talukder, K .H and Harada, K. " Enhancement of Discrete Wavelet Transform (DWT) for Image Transmission over Internet", *Eighth International Conference on Information Technology: new generations*, 2011.

[6] Y.lano, D.S. Silvestre da Silva, and Ana Lucia Mendes Cruz, "A Fast and efficient Hybrid Fractal-Wavelet Image Coder," *IEEE Trans. on Image Processing*, Vol. 15, No. 1, 2006

[7] D. Duh, J. Jeng, S. Chen. "DCT based simple classification scheme for fractal image compression", *Image and Vision Computing*, 2005, 23(13): 1115–1121.

[8] Chuanwei, S., Quanbin, L. and Jingao, L. "The Study of Digital Image Compression Based on Wavelets ", 2010.

[9] Lotfi A. A., Hazrati M. M., Sharei M., Saeb Azhang, " Wavelet Lossy Image Compression on Primitive FPGA", *IEEE*, pp. 445-448, 2005.

[10] Vinay U. Kale & Nikkoo N. Khalsa , "Performance Evaluation of Various Wavelets for Image Compression of Natural and Artificial Images", *International Journal of Computer Science & Communication,* Vol. 1, No. 1, January-June 2010, pp. 179-184.

[11] Gohar Vahdati, Habib Khodadadi, Mahdi and Mohammad-R. Akbarzadeh-T, "Fractal Image Compression Based on Spatial Correlation and Hybrid Particle Swarm Optimization with Genetic Algorithm", *In Proceedings of the International Conference on Software Technology and Engineering*, pp.185-189, 2010.

[12] R. C. Eberhart and Y. H. Shi, "Particle swarm optimization: Developments, applications and resources," *in Proc. IEEE Int. Congr. Evolutionary Computation , Seoul, Korea* , 2001, vol. 1, pp. 81– 86.

[13] C.A.C. Coello, G.T. Pulido, and M.S. Lechuga . "Handling multiple objectives with particle swarm optimization" . *IEEE Transactions on Evolutionary Computation*, 8(3):256{279, June 2004.