Iteration Free Fractal Image Compression For Color Images Using Vector Quantization, Genetic Algorithm And Simulated Annealing

A R Nadira Banu Kamal

Dept of Computer Science, TBAK College for Women, Kilakarai, Tamil Nadu, India
nadirakamal@gmail.com

Abstract: This research paper on iteration free fractal image compression for color images using the techniques Vector Quantization, Genetic Algorithm and Simulated Annealing is proposed, for lossy compression, to improve the decoded image quality, compression ratio and reduction in coding time. Fractal coding consists of the representation of image blocks through the contractive transformation coefficients, using the self-similarity concept present in the larger domain blocks. Fractal coding achieves high compression ratio but it consumes more time to compress and decompress an image. Different techniques are available to reduce the time consumption and improve the decoded image reliability. But most of them lead to a bad image quality, or a lower compression ratio. Usage of synthetic codebook for encoding using Fractal does not require iteration at decoding and the coding error is determined immediately at the encoder. The techniques Vector Quantization, Genetic Algorithm and Simulated Annealing are used to determine the best domain block that matches the range blocks. The proposed algorithm has the better performance in terms of image quality, bit rate and coding time for Color images. Only the encoding consumes more time but the decoding is very fast.

Key words: Fractal code, Iteration free, Vector quantization, Genetic algorithm, Simulated annealing

Introduction

With the recent rapid growth of multimedia applications and digital transmission, image compression techniques have become a very important subject. A digital image obtained by sampling and quantizing a continuous tone picture requires an enormous storage. For instance, a 24 bit color image with 512x512 pixels will occupy 768 Kbytes storage on a disk and a picture twice of this size will not fit in a single floppy disk. To transmit such an image over a 28.8 Kbps modem would take almost 4 minutes (Vivek Arya et al., 2013). Thus image compression addresses the problem of reducing the amount of data required to represent a digital image, so that it can be stored and transmitted more efficiently. Fractal compression is one of the methods and this method is used in this research work to achieve improved image quality, compression ratio and reduction in coding time. The proposed method has the advantages such as low time consumption for decoding and less memory requirements for storage, which is most needed in today's communication.

Fractal coding consists of the representation of image blocks through the contractive transformation coefficients, using the self-similarity concept present in the larger domain blocks. Fractal coding achieves high compression ratio but the time required to compress and decompress an image is time consuming. There are many modified versions proposed to improve the fractal coding techniques (Ghazel et al., 2005, Mohsen et al., 2003). Most of the studies focus on

- refining the block transformation (Chong and Minghong, 2001)
- reducing of the complexity of the encoding process (Chen et al., 2002)
- speeding up the process (Hau-Jie and Wang, 2005, Riccardo et al., 2006)

An iteration-free fractal image coding for color images using the techniques Vector Quantization (VQ), Genetic Algorithm (GA) and Simulated Annealing (SA) is proposed for lossy compression in this research work to improve decoded image quality, compression ratio and to reduce the coding time. The major problems with a VQ encoder are the codebook search complexity and the memory required to store the codebook. Both the codebook search complexity and storage requirements increase exponentially with the vector dimension (Jeng et al., 2003). Therefore the size of the VQ blocks is usually kept very small which in turn results in a high statistical correlation between adjacent blocks. In this research work the size of the vector is limited to 64 ie., 8x8 blocks. The searching process of the reproduction vector for an input vector using full search requires intensive computations. Many algorithms are proposed to speed up the searching process. The algorithm given by Jim and Yi (2004) uses the vector's features (mean value, edge strength, and texture strength) to delete impossible codewords that cannot be rejected by the DHSS algorithm using three projections. Two additional inequalities, one for terminating the searching process and another to reject impossible codewords, were presented to reduce the distortion computations. This algorithm is better than the DHSS algorithm and Pan's method (Pan et al., 2003) in terms of computing time and the number of distortion calculations hence this

algorithm was included in the present research work for eliminating the domain blocks while searching for the optimal solution in the proposed method using the technique VQ. As genetic algorithms and Simulated Annealing are global search and optimization method, it can be applied to the proposed method to find the optimal domain block for each range block in iteration-free fractal color image coding. The problems faced in GA are encoding the solution, fixing the size of the population, choosing the crossover and its probability, mutation and its probability and selection procedure. Similarly the problems faced in SA are generating a candidate solution, problem-specific cooling schedule, objective function and Metropolis-step in which the algorithm decides if a solution is accepted or rejected. The selection of these parameters greatly affects the solution. The proposed method using the VQ technique aims to design an efficient domain pool for the fractal coding schemes and it also introduces the optimization techniques like GA and SA to improve the quality of the decoded image and for speeding the encoding time in generating the fractal codes. This research work of introducing VQ, GA and SA reduces the computational complexity, improves the image quality and reduces the coding time.

Algorithm of the proposed method

Usage of synthetic codebook for encoding using Fractal does not require iteration at decoding and the coding error is determined immediately at the encoder (Chang and Chung, 2000). Hence there is a reduction in decoding time. In the proposed method a synthetic codebook is created as the domain pool using the mean image, whose pixel values are the block means of all the range blocks. This code book is used as the domain pool for genetic algorithm and simulated annealing techniques. As these techniques have become an efficient tool for search and optimization, they can be used to find the better domain block from the domain pool that matches the range block of the image in fractal coding.

Parameters for VQ

In the iteration-free fractal image coding using vector quantization the domain pool was classified and pruned for each range block for each RGB component before the comparison for the identification of the better domain block from the codebook for the range block in order to reduce the computational time (Jim and Yi, 2004). The proposed methodology using the VQ technique reduces the coding process time and intensive computation tasks by pruning the domain block for each range block. The redundancies in the domain pool are first reduced by the Linde Buzo Gray (LBG) Algorithm. Further redundancy in the domain block for each range block was achieved using the vector features such as mean value, edge strength, and texture strength (Nadira and Priyanga, 2014a). A pruning condition for terminating the searching process to find the best domain block from the domain pool has been used in this proposed research work.

Parameters of GA

Crossover is made in hope that new chromosomes will contain good parts of old chromosomes and therefore the new chromosomes will be better. However, it is good to let some part of old populations survive to next generation. The crossover probability is chosen as 0.85 in the proposed method. Mutation generally prevents the GA from falling into local extremes. Mutation should not occur very often, because GA will in fact then change to random search. The mutation probability is chosen to be 0.06 in the proposed method (Nadira and Priyanga, 2014). Research shows that after some limit (which depends mainly on encoding and the problem) it is not useful to use very large populations because it does not solve the problem faster than moderate sized populations. In the proposed method the size of the population is chosen to be 40. There are many methods in selecting the best chromosomes. Examples are Roulette wheel selection, Boltzmann selection, Tournament selection, Rank selection, Steady state selection and some others. Roulette wheel selection is used in the proposed method.

Parameters of SA

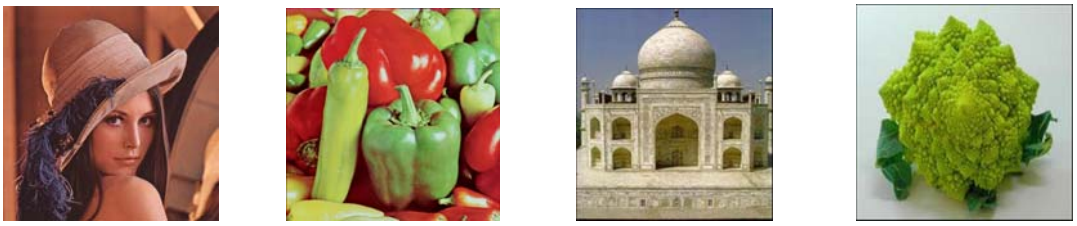
At each step, the SA heuristic considers some neighbors of the current state **s**, and probabilistically decides between moving the system to state **s'** or staying back in state **s**. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Boltzmann's Probability is used in the proposed method. In the present problem the neighbor of a state is any domain block from the domain pool. The probability of making the transition to the new state **s'** is a function P(δE, T) of the energy difference δE = E(s') - E(s) between the two states, and of a global time-varying parameter T called the control parameter (Nadira , 2012). In the proposed method T is made to vary from 1 to 0. Initially, T is set to a high value (or infinity), and it is decreased at each step according to some annealing schedule. Various cooling schedules are suggested in

literature. In the proposed method $T_i = T_0 - i\frac{T_0 - T_{N2}}{N2}$ is used as it leads to better solution (Nadira and Thamaraiselvi, 2006a).

The sender sends the color image for compression. In the preprocessing stage, the input MxNx3 image under coding is divided into non-overlapping square blocks of BxBx3 pixels called the range blocks. Then the mean and variance of each range blocks are determined. After the mean of all the range blocks are obtained, a mean image of size M/B x N/B x 3 with each pixel corresponding to the block mean is generated. The mean image must be larger than the size of the range block i.e. M/B x N/B x 3 > B x B x 3. The maximum size of B is limited to 8 in order to produce a good quality of the decoded image. The higher the resolution of the input image (MxNx3) more blocks can be generated for the domain pool which helps to find a good mapping between the domain and range blocks. The initial domain pool with blocks of the same size as the range is generated using the mean image. In the encoder if the variance of the range block is smaller than the threshold value *E*, the range block is coded by the mean, or else the range block will be coded by the contractive affine transformation (Chang, 2001). The aim of the proposed scheme is to find the domain block for each image range block and the transformation parameters that minimize the distortion between the image block and the transformed domain block in a minimized time. This process of finding the best domain block makes use of the techniques like VQ, GA and SA. In the decoder, the mean information of each range block is extracted from the fractal codes. Using this information the mean image is constructed. This mean image is partitioned into blocks of the same size as the input image. This forms the domain pool for GA and SA search methods but for VQ the domain pool is constructed from the mean image blocks (same size as that of the input) using LBG algorithm. The decompressed image is constructed block by block by applying the transformation parameters to the selected domain block from the domain pool as per the code.

Implementation

For implementation of the proposed algorithms, four 512 x 512 benchmark color images of Lena, Pepper, Tajmahal and Cauliflower [shown in Figure 1 (a) to (d)] with twenty four bit color resolution were used. In the simulation, the images were partitioned into range blocks with the block size, 8x8 or 4x4 or 2x2. The maximum block size is set to 8x8 because for a range block size greater than 8x8 the determination of the proper domain block was difficult and the quality of the image reconstructed was poor. The threshold value for the variance of range blocks was chosen by trial and error basis to be of size 20 for block size 8x8, 10 for 4x4 and 5 for 2x2 that results in good compression ratio and PSNR. The number of blocks in the mean image is the size of the domain pool. These algorithms were implemented using the software Matlab 7.12 on the Intel (R) Core[TM] 2 E7500 systems with 2.93 GHz and 1.96 GB of RAM.



(a) Lena (b) Pepper (c) Tajmahal (d) Cauliflower

Figure 1: Original (512 X 512x3, 24 Bit/Pixel) Images.

The range block with a size 8x8, 4x4 and 2x2 was considered for simulation. The length of the attached header to the proposed iteration-free fractal code for each range block was one bit because it only denoted whether or not the range block was coded by the mean. For an image partitioned by 4x4 range blocks, every block mean was calculated and a 128x128 mean image was obtained. Figure 2 (b) shows the mean image of Lena got by this partition and it is very similar to its original image except its size. Therefore the domain pools of different sizes namely 16, 32 and 64 using the LBG-based method from the mean image was constructed for VQ. For GA and SA the range block with a size 8x8, 4x4 & 2x2 was considered for simulation. Here the total number of range blocks for the block size 4x4 for each color component was n = 16384 and total number of domain blocks (m) to search were (128 / 4) x (128 / 4) = 32 x 32. Thus, the cardinality (N1) of the search spaces for this case was 8 x 4 x 1024.The string length n was taken to be 15 (3 + 2 + 10). In GA out of these 2^{15} binary strings, forty strings (S = 40) were selected randomly to construct an initial population. A high crossover probability, say $p_c = 0.85$, was taken for the crossover operation. For mutation operation, p_m was 0.06. Roulette-wheel selection procedure

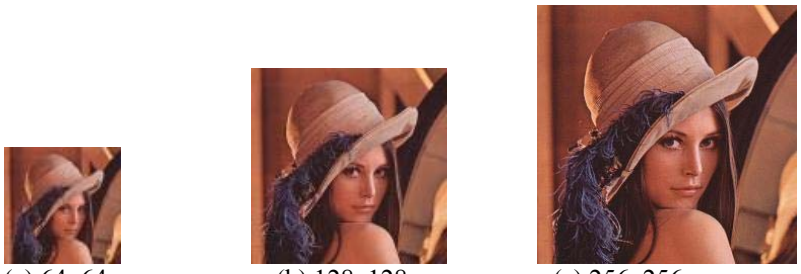
was used. The fitness value of a string between the given range block and the obtained range block for each RGB color component is taken to be the MSE given in Eq. 1.

$$MSE\ (R, \hat{R}) = \frac{1}{B^2} \sum_{0,i,j \le B} \left(r_{i,j} - \hat{r}_{i,j} \right)^2 \qquad (1)$$

The probability of selection of a string in the population to the mating pool was inversely proportional to its fitness value because the present optimization problem is a minimization problem. The total number of generations (iterations) considered in the GA was T = 60. Hence, the search space reduction ratio was approximately 14. For the block size 8x8, the total number of range blocks for each RGB component was n = 4096 and total number of domain blocks to search was (64 / 8) x (64 / 8) = 8 x 8. Thus, the cardinality (N1) of the search spaces for this case was 8 x 4 x 64. The string length n was taken to be 11 (3 + 2 + 6). For the block size 2x2, the total number of range blocks was n = 65536 and total number of domain blocks to search was (256 / 2) x (256 / 2) = 128 x 128. Thus, the cardinality (N1) of the search spaces for this case was 8 x 4 x 16384.The string length n was taken to be 19 (3 + 2 + 14). The domain pool was found and the coding performance using the above parameters was simulated. Using SA for the block of size 8x8, 4x4 and 2x2 the solution configuration length was taken to be 11, 15 and 19 respectively. Knuth random numbers were globally generated for the rearrangement of the solution so that most possible solutions were considered. The seed value was chosen to be as 0.5. In the simplest form of the problem, the objective function *was* taken as the Mean Square Error (MSE). Some random rearrangement of the solution was first generated, and used them to determine the range of values the objective function encountered from move to move. Choosing a starting value for the parameter T_0 which was considerably large say 1, and process was conducted downward each amounting to a decrease in *T*. Each new value of *T* was kept constant for, say, *N* (N=40) reconfigurations and store the configuration which produces minimum value for the objective function. If the terminating condition was not reached, the parameter for *T* was reduced and the next trial was started with the configuration that produced minimum value for the objective function. The equation for the parameter *T* was chosen as

$$T_i = T_o - i \frac{T_o - T_N}{N} \qquad (2)$$

where T_o = 1 and T_N = 0 and *i* is the iteration number. The coding performance with the contractive affine transformation under the different sizes for the domain pool on the parameters like coding time, image quality and bit rate was determined.



(a) 64x64 (b) 128x128 (c) 256x256

Figure 2: Mean Images Of Lena.

Results and Discussions

The range blocks were classified before coding [Yung et al., 2003]. Range blocks were grouped into two sets according to the variability of the pixel values in these blocks. If the variability of a block was low, i.e., if the variance of the pixel values in the block was below a fixed value, called the threshold, the block is called smooth type range block. Otherwise, it is called a rough type range block.

Table 1: Classification of blocks and bit rate using different types of encoding on the chosen images using the proposed techniques.

Image	Range Block size	No of Range Blocks for GA/SA		Bit Rate		Compression ratio		PSNR		
				GA / SA	VQ	GA / SA	VQ			
		Smooth	Rough					GA	SA	VQ
Lena	2 * 2	43927	21611	7.56	7.47	3.17	3.20	41.60	35.41	31.55
	4 * 4	8394	7992	1.95	2.04	12.26	12	33.12	31.68	27.84
	8 * 8	1559	2539	0.48	0.54	49.82	44.50	28.76	28.36	25.16
Pepper	2 * 2	51409	14129	7.02	7.16	3.41	3.35	39.84	36.24	30.78
	4 * 4	9851	6535	1.87	1.99	12.81	12.06	33.52	32.20	26.92
	8 * 8	1638	2460	0.47	0.53	50.16	45.19	27.84	27.84	23.91
Cauliflower	2 * 2	49433	16105	7.16	7.04	3.34	3.40	37.63	37.44	27.03
	4 * 4	8640	7746	1.94	1.87	12.34	12.77	29.49	28.31	23.45
	8 * 8	1682	2416	0.47	0.49	50.36	48.87	24.23	24.88	21.97
Tajmahal	2 * 2	44672	20866	7.51	7.35	3.19	3.26	42.83	39.90	22.50
	4 * 4	6624	9762	2.05	1.98	11.65	12.07	33.86	32.32	25.49
	8 * 8	1018	3080	0.50	0.57	47.58	44.44	26.64	26.52	30.42

The purpose of choosing this block classification was for two reasons. One is to get higher compression ratio, and the other is to reduce the coding time. The threshold value that separates the range blocks into two types was chosen as stated earlier. After classification, VQ-based, GA-based and SA-based coding was adopted for the rough type range blocks only. All the pixel values in a smooth type range block were replaced by the mean of its pixel values. [Table 1] gives the classification of blocks and bit rate using different types of encoding using the proposed techniques on the images chosen for simulation. From the results tabulated in [Table 1], it is observed that for the images which have the number of smooth blocks significantly high has a high compression ratio.

Figure. 3. Decompressed Images of Lena, Pepper, Cauliflower and Tajmahal using VQ for block size 8x8 using 16 level

Figure 4: Decompressed Images of Lena, Pepper, Cauliflower and Tajmahal using VQ for block size 4x4 using 32 level

The decompressed image of Lena, Pepper, Cauliflower and Tajmahal for the block partition of sizes 8 x8 using 16 level, 4x4 using 32 level and 2x2 using 64 level for the technique VQ is shown in Figure 3,4and 5 respectively.

Figure 5: Decompressed Images of Lena, Pepper, Cauliflower and Tajmahal using VQ for block size 2x2 using 64 level

(a) Block size 8x8 (b) Block size 4x4 (c) Block size 2x2

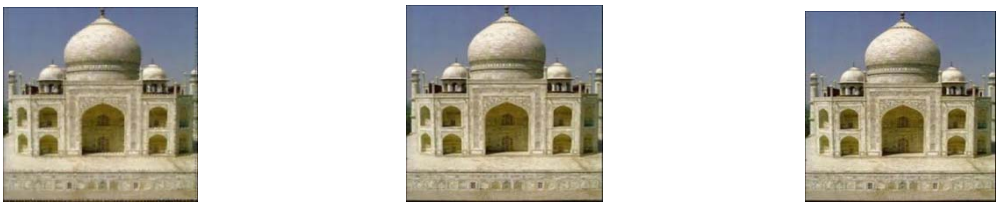
Figure 6 : Decompressed Image Of Lena For Block Size 8x8, 4x4 And 2x2 Using The Proposed GA Technique.

(a) Block size 8x8 (b) Block size 4x4 (c) Block size 2x2

Figure 7: Decompressed Image Of Cauliflower For Block Size 8x8, 4x4 And 2x2 Using The Proposed GA Technique.

(a) Block size 8x8 (b) Block size 4x4 (c) Block size 2x2

Figure 8: Decompressed Image Of Pepper For Block Size 8x8, 4x4 And 2x2 Using The Proposed GA Technique.



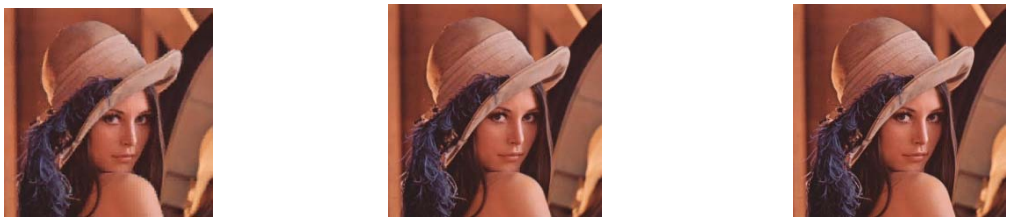
(a) Block size 8x8 (b) Block size 4x4 (c) Block size 2x2

Figure 9: Decompressed Image Of Tajmahal For Block Size 8x8, 4x4 And 2x2 Using The Proposed GA Technique.

The decompressed image of Lena, Cauliflower, Pepper and Tajmahal for the block partition of sizes 8 x8, 4x4 and 2x2 using the technique GA is shown in Figure 6,7,8 and 9 respectively. The decompressed image of Cauliflower, Lena, Pepper and Tajmahal for the block partition of sizes 8 x8, 4x4 and 2x2 using the technique SA is shown in Figure 10,11,12 and 13 respectively.

(a) Block size 8x8 (b) Block size 4x4 (c) Block size 2x2

Figure 10: Decompressed Cauliflower Image for Block Size 8x8, 4x4 And 2x2 Using the Proposed SA Technique



(a) Block size 8x8 (b) Block size 4x4 (c) Block size 2x2

Figure 11: Decompressed Lena Image for Block Size 8x8, 4x4 And 2x2 Using the Proposed SA Technique

(a) Block size 8x8 (b) Block size 4x4 (c) Block size 2x2

Figure 12: Decompressed Pepper Image for Block Size 8x8, 4x4 And 2x2 Using the Proposed SA Technique

(a) Block size 8x8 (b) Block size 4x4 (c) Block size 2x2

Figure 13: Decompressed Tajmahal Image for Block Size 8x8, 4x4 And 2x2 Using the Proposed SA Technique

The Encoding time of the proposed techniques using GA, SA and VQ values are reported in the [Table 2]. These results of encoding time using block sizes 8x8, 4x4, 2x2 are plotted in the graph and shown in Figure 14 and the and their PSNR values are plotted in the graph shown in figure 15.

Table 2: Results of the Proposed Iteration-Free Fractal Method Using the Techniques VQ, GA and SA

Image	Range Block Size	Using GA		Using SA		Using VQ					
		RMS	Coding Time (secs)	RMS	Coding Time (secs)	Codebook size 16		Codebook size 32		Codebook size 64	
						RMS	Coding Time (secs)	RMS	Coding Time (secs)	RMS	Coding Time (secs)
Lena	8x8	9.29	31250	9.64	17784	14.06	5941.4	14.30	6374.3	14.29	7009.6
	4x4	5.62	63871	6.64	51921	10.33	23272	10.14	70142	10.19	31386
	2x2	2.12	110330	4.32	106660	6.74	79228	6.69	104390	6.57	153450
Pepper	8x8	10.84	36276	10.33	17076	16.24	5855	16.38	6756.4	16.57	6881
	4x4	5.37	52157	6.25	22802	11.50	21268	11.23	24023	11.35	28241
	2x2	2.59	77985	3.92	70342	7.37	55990	7.30	160930	7.23	109640
Cauliflower	8x8	15.39	34054	14.52	17743	20.31	4562.8	20.29	4789.2	20.34	5265.7
	4x4	8.54	50683	9.78	20484	17.12	16334	16.92	18459	16.77	22585
	2x2	3.34	84643	3.42	64120	11.34	49001	11.31	66905	11.25	96249
Tajmahal	8x8	11.86	31250	12.02	28842	19.11	8817.3	19.49	5616.4	19.52	6408.1
	4x4	5.16	64737	6.16	17794	13.53	14163	13.39	24725	13.39	31149
	2x2	1.84	110470	2.57	60320	7.68	38014	7.52	95678	7.39	138210

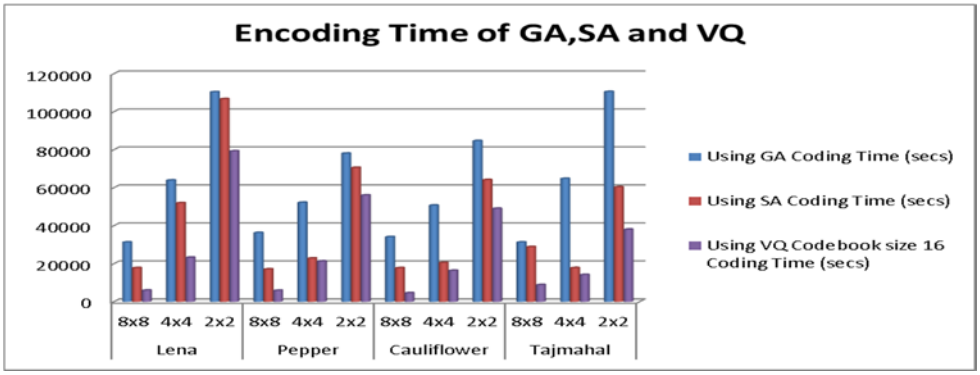


Figure 14: Encoding Time of GA, SA and VQ

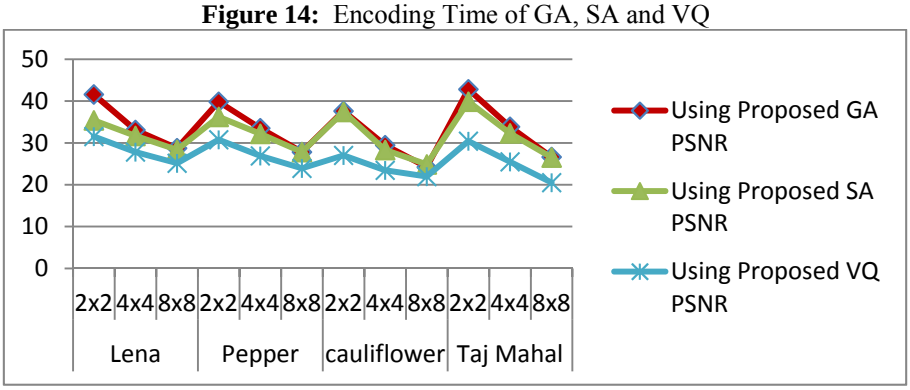


Figure 15: PSNR Of Lena, Pepper, Cauliflower and Tajmahal for Block Size 8x8, 4x4 and 2x2 Using the Technique GA, SA And VQ.

From the results obtained, it is obviously clear that GA technique for iteration-free fractal coding is preferred for better image quality whereas VQ is preferred for reduced coding time and SA is preferable for optimal image quality and time. The proposed methods using VQ, GA and SA are found to provide computational efficiency, thereby drastically reducing the cost of coding. [Table 3] gives the result of similar methods found in the literature and the proposed methods for the bench mark image of Lena and Pepper (512 x 512, 24 bit color image). In the proposed method using SA, GA and VQ the PSNR is highly effective when compared to the existing fractal methods given in references (Somasundaram, 2011) and (Nileshsingh and Kakde, 2007)

Table 3: PSNR of some methods for 512x512 color images

Image	Range	SA	GA	VQ	RGB & Gray Scale Component On MPQ-BTC In Image Compression (Somasundaram, 2011)	Color Image Compression with Modified Fractal Coding on Spiral Architecture (Nileshsingh and Kakde, 2007)
Lena	4 x 4	31.68	33.12	27.84	24.1209	29.05
Pepper	4 x 4	32.20	33.52	26.92	24.1531	-

Conclusion

In this Work, a fast-encoding algorithm for fractal image coding is proposed and implemented using Genetic Algorithm, Simulated Annealing and Vector Quantization for still color images. First the range blocks were classified as either smooth or rough depending on the variance of the block. This classification was very useful when the image had lot of smooth blocks. So depending on the image and the partition, a high compression ratio was achieved. Only the encoding consumes more time but the decoding is very fast. GA technique for iteration-free fractal coding is preferred for better image quality whereas VQ is preferred for reduced coding time and SA is preferable for optimal image quality and time. The proposed methods using VQ, GA and SA are found to provide computational efficiency, thereby drastically reducing the cost of coding. The execution time can further be reduced by implementing the proposed method in parallel for encoding. Color images are commonly used in most of the application now-a-days. Applications where images can be stored in a compressed form, which require faster retrieval, like medical images and photographs for identification can use the proposed method.

Acknowledgment

This research work is supported by the UGC, New Delhi, India

References

Chang and Chung, (2000) .Hsuan T. Chang and Chung J. Kuo. Iteration-Free Fractal Image Coding Based on Efficient Domain Pool Design. IEEE Trans. Image Processing, **9(3)**, 329-339, Mar 2000.

Chang, (2001). Hsuan T. Chang. Variance-Based Domain Blocks for Non-Iterative Fractal Image Compression. Journal of Chinese Institute of Electrical Engineering (EI), **8(2)**, 159-169, April 2001.

Chen et al., (2002). Chen Yisong; Wang Guoping; Dong Shihai. Feature Difference Classification Method in Fractal Image Coding. Proceedings of the 6th International Conference on Signal Processing, **1**, 26-30 Aug. 2002, 648 – 651, 2002.

Chong and Minghong, (2001). Chong Sze Tong and Minghong Pi. Fast Fractal Image Encoding Based on Adaptive Search. IEEE Transactions on Image Processing, **10(9)**, 1269-1277, September 2001.

Ghazel et al.,(2005). M.Ghazel, R. K. Ward, R. Abugharbieh,E. R. Vrscay, G. H. Freeman. Simultaneous Fractal Image Denoising and Interpolation. 0-7803-9195-0/05/$20.00 ©2005 IEEE, 558-561, 2005.

Hau-Jie and Wang, (2005). Hau-Jie Liang and Shuenn-Shyang Wang. Architectural Design Of Fractal Image Coder Based On Kick-Out Condition. 0-7803-8834-8/05 $20.00 © 2005 IEEE, 1118-1121, 2005.

Jeng et al.,(2003). Jeng -Shyang Pan, Zhe-Ming Lu and Sheng-He Sun. An Efficient Encoding Algorithm for Vector Quantization Based on Subvector Technique. IEEE Transactions on Image Processing. **12(3)**, 265-270, March 2003.

Jim and Yi,(2004) .Jim Z. C. Lai and Yi-Ching Liaw. Fast-Searching Algorithm for Vector Quantization Using Projection and Triangular Inequality. IEEE Transactions on Image Processing, **13(12)**, 1554-1558, December 2004.

Mohsen et al., (2003). Mohsen Ghazel, George H.Freeman and Edward R.Vrscay. Fractal Image Denoising, IEEE Trans. Image Processing, **12(12)**, 1560–1578, Dec. 2003.

Nadira and Thamaraiselvi, (2006a) .A.R.Nadira Banu Kamal and Dr S. Thamarai Selvi. Iteration-Free Fractal Coding For Image Compression Using Simulated Annealing. Proceedings of the IET International Conference on Visual Information Engineering 2006, Institute of Engineering and Technology, United Kingdom, 189-194, Sep 2006.

Nadira ,(2012). A.R.Nadira Banu Kamal, "Iteration Free Fractal Compression Using Simulated Annealing for still Colour Images", UACEE International Journal of Artificial Intelligence and Neural Networks,**2(3),**10-14,Dec 2012.

Nadira and Priyanga, (2014). A.R.Nadira Banu Kamal and P.Priyanga, "Iteration Free Fractal Compression Using Genetic Algorithm for still Colour Images", ICTACT Journal on Image and Video Processing,**4(3),** 785-790, Feb 2014.

Nadira and Priyanga, (2014a). A.R.Nadira Banu Kamal and P.Priyanga, "Iteration Free Fractal Color Image Compression Using Vector Quantization", International Journal of Advanced Research in Computer and Communication Engineering, **3(1),** 5154-5163, Jan 2014.

Nileshsingh and Kakde, (2007) .Nileshsingh V. Thakur and Dr.O.G Kakde, "Color Image Compression with Modified Fractal Coding on Spiral Architecture" Journal of Multimedia, **2(4):** 55-66, Aug 2007.

Pan et al.,(2003) .T.S.Pan, Z. M. Lu and S.H. Sun. An Efficient Encoding Algorithm for Vector Quantization Based on Subvector Techniques, IEEE Trans. Image Processing **12(3),** 265 – 270, 2003.

Riccardo et al., (2006) .Riccardo Distasi, Michele Nappi, and Daniel Riccio. A Range/Domain Approximation Error-Based Approach for Fractal Image Compression, IEEE Transactions on Image Processing, **15(1)**, 89- 97**,** January 2006.

Somasundaram ,(2011). K. Somasundaram and P.Sumitra, "RGB & Gray scale Component on MPQ-BTC in Image Compression", International Journal on Computer Science and Engineering,.**3(4):**1462-1467, Apr 2011.

Vivek Arya et al., (2013). Vivek Arya, Dr.Priti Singh, Karamjit Sekhon, RGB Image compression using Two Dimensional Discrete Cosine Transform, International Journal of Engineering Trends and Technology, **4(4),**828-832,April 2013.

Yung et al., (2003) .Yung -Gi, Wu, Ming-Zhi, Huang, Yu-Ling, Wen. Fractal Image Compression with Variance and Mean, ICME 2003, I- 353 to I-356, 2003.