Memetic Algorithm Based Vector Quantizer

Mrs.V.S.Jayanthi[1], Dr.A.Shanmugam[2]

[1]Research Scholar, Dept. of ECE, P S G College of Technology, Coimbatore,India.

[2]The Principal, Bannariamman Institute of Technology, Sathyamanagalam,India.
vsj1965@gmail.com

Abstract

Image compression is the key technology in the development of various multimedia applications Vector quantization is a universal and powerful technique to compress a data sequence, such as speech or image, resulting in some loss of information In VQ, minimization of Mean Square Error (MSE) between code book vectors and training vectors is a non-linear problem. Traditional LBG types of algorithms used for designing the codebooks for Vector Quantizer converge to a local minimum, which depends on the initial code book. Memetic algorithms (MAs) are population-based meta-heuristic search approaches that have been receiving increasing attention in the recent years. These algorithms are inspired by models of natural systems that combine the evolutionary adaptation of a population with individual learning within the lifetimes of its members. It has shown to be successful and popular for solving optimization problems. In this paper we present a new approach to vector quantization based on memetic algorithm. Simulations indicate that vector quantization based on memetic algorithm has better performance in designing the optimal codebook for Vector Quantizer than conventional LBG algorithm. The Peak Signal to Noise Ratio (PSNR) is used as an objective measure of reconstructed image quality.

Keywords: *Vector quantization, Memetic algorithms, hybrid GAs, The global optimum*

I. INTRODUCTION

Multimedia has become a vital part of a diverse range of areas such as engineering, finance, medical sciences and entertainment in recent years. Typical multimedia signals include audio, video, speech, image, medical, and musical. The rapid evolution of multimedia services coupled with the phenomenal increase in the PC computational power and network bandwidth capacity have posed several new challenges in the area of multimedia data processing and compression. There exists a continuing need to find more efficient techniques and algorithms for manipulation, modeling, analysis, implementation and compression of multimedia data

Image compression algorithms using Vector Quantization (VQ) have been receiving considerable attention. Vector quantization [1], a compression technique which is a multidimensional extension of scalar quantization, partition the input image to be encoded into two dimensional vectors, each of which is compared by an encoder with every code vector in a previously designed codebook. The index of the code vector which best matches the input block according to some distortion metric is sent to the decoder, which uses the index to look up reconstruction vector in its copy of the codebook. The reconstruction vectors are tiled to form a lossy compressed version of the input image.

Vector quantization [1], [4] plays an important role in data compression and it has been successfully used in speech compression and image compression. A vector Quantizer (VQ) Q of dimension k can be defined as a mapping of data vectors X in k-dimensional Euclidean space R^k, into a finite subset Y of R^k. Let X be a set of training vectors of size M and dimension k, that is., $X=\{x_1, x_2 .. x_M\}$, $x_i \in R^k \ \forall\ i = \{1,2, ... , M\}$. Let Y be a set of code words of size N and dimension k that is. $Y = \{y_1, y_2 , .. y_N\}$, $y_i \in R^k, \forall\ i =\{1,2, .. ,N\}$. A data vector $x \in R^k$ is encoded by identifying the index j of the codevector $y_j \in Y$ such that $\| x- y_j \| \leq \| x- y_i \| \ \forall\ i \neq j$. The decoder uses the received index j to retrieve the code word from the codebook and generates the reconstruction vector y_j corresponding to x. The distortion measure used is mean square error (MSE) given by $d(x, y_j) = \| x- y_j \|^2$. If a VQ minimizes the average distortion, it is called the optimal VQ of size N.

The efficiency of a Vector Quantizer mainly depends on the codebook used. Several algorithms have been proposed to design a codebook using the information provided by the training vector set and is based upon the minimization of the distortion measure. The most widely used is LBG algorithm [1], proposed by Linde, Buzo and Gray also known as the k- means algorithm. LBG algorithm is an iterative gradient descent algorithm that tries to minimize an average squared error distortion measure. It starts with an initial solution, which can be chosen arbitrarily. The existing solution is then improved iteratively using the optimality criteria in turn until a minimum is reached. The algorithm is relatively easy to implement and it gives reasonable results in most cases. Unfortunately the algorithm makes only local changes to the original codebook and it settles with the first local minimum. The quality of the final codebook therefore highly depends on the initialization.

In this paper, our key focus is to investigate the application of memetic algorithms to the design of codebook for Vector Quantization. This paper is organized as follows: Section II describes and outlines the memetic algorithms. In section III memetic algorithm based vector quantizer is presented. Section IV discusses the experimental results. Finally Section V gives the conclusion.

II. MEMETIC ALGORITHM

Genetic algorithms (GAs) [2], [3] are a powerful set of search and optimization techniques that are capable of exploring and exploiting promising regions of the search space. They can, however, take a relatively long time to locate the local optimum in a region of convergence and in some cases may not find the global optimum. Torn and Zilinskas [5], entitled Global search methods: exploration and exploitation, observe that two competing goals govern the design of global search methods: exploration is important to ensure global reliability; i.e., every part of the domain is searched enough to provide a reliable estimate of the global optimum; exploitation is also important since it concentrates the search effort around the best solutions found so far by searching their neighborhoods to produce better solutions. Any carefully designed GA is only able to balance the exploration and the exploitation of the search effort, which means that an increase in the accuracy of a solution can only come at the sacrifice of convergent speed, and vice visa. It is unlikely that both of them can be improved simultaneously.

Genetic algorithms (GAs) are based on the principle of Darwinian Evolution. In GA, the solutions to a given problem are encoded in chromosomes. GAs use the principle of biological evolution to generate successively better solutions (chromosomes) from previous generations of solutions, by applying the three basic genetic operators, crossover, mutation and natural selection. Genetic algorithms (GAs) perform well for global searching, but they generally suffer from excessively slow convergence to locate a precise solution because of their failure to exploit local information. Hence pure Genetic algorithms are not well suited to fine tuning search in complex combinatorial spaces. On the other hand, local search methods can quickly find the local optimum of a small region of the search space, but are typically poor global searchers. Therefore, the Genetic Algorithms (GAs) can be hybridized with the local search methods for a particular problem to improve their performance. Such hybrids GAs are often called Memetic Algorithms (MAs)[3]. The term memetic algorithm was introduced by Moscato and Norman ('92) to describe Evolutionary Algorithms in which local search plays a significant part. The term was motivated by Dawkin's notion ('76) of a meme as a unit of information that reproduces itself as people exchange ideas.

MAs[6] are inspired by Neo-Darwinian's principles of natural evolution and Dawkins' notion of a meme defined as a unit of cultural evolution that is capable of local refinements. Memetic algorithm (MA) can be regarded as an extension of GA that incorporates a local-search algorithm for the solution in between generations. Local search is performed to improve the fitness of the population (in a localized region of the solution space) so that the next generation has better genes from its parents. Hence convergence time of Memetic algorithm is also reduced. Memetic Algorithms incorporate the concept of memes by allowing individuals to change before the next population is produced. Individuals may copy parts of genes from other individuals to improve their own fitness. Hence it can be said that MA uses the local search methods to serve the genetic operators with solutions those are better in comparison to randomly generated solutions. The local search algorithm adopted in a Memetic Algorithm is somewhat dependent on the problem being solved.

The process of performing a local search can be thought of as a consequence of individual learning during the lifetime of the individual. N. Shahidi, H. Esmaeilzadeh, M. Abdollahi, and C. Lucas, has applied Memetic Algorithm to the problem of finding the optimal collision free path for a mobile robot [7]. In [8],J.E.Smith has investigated about Co-evolving memetic algorithms. The basic steps of the MAs are outlined in Fig 1.

In this paper, the application of memetic algorithms to the design of codebook for Vector Quantization for compressing the image is proposed. The proposed scheme is developed in such a way that a simple GA is acting as a base level search, which makes a quick decision to direct the search towards the optimal region, and a local search method is next employed to do the fine tuning. The proposed technique can outperform conventional genetic algorithms in the sense in that MAs make it possible to improve both the quality of the solution and reduce the computing expenses.

```
Procedure:: Memetic Algorithm
Begin
Initialize: Generate an Initial Population of individuals
Perform Local search
Evaluate all individuals in the population
While (terminating conditions are not satisfied)
  Select individuals from population based on their fitness criteria to create a mating pool
   Crossover: Randomly pair individuals (i.e. parents) from the mating pool and apply the crossover
   operator producing two off- springs from each pair. Newly created off-springs are added to the off-
   spring set.
   Mutation: Apply with low probability the mutation operator to the offspring set.
   Perform Local search
    Evaluate all off-springs in the population and replace the least fit parent chromosomes in the
   existing population by the newly generated off-springs
    End While
End
```

Figure 1: Pseudo code of a standard Memetic Algorithm

III. MEMETIC VECTOR QUANTIZER

The experimental steps of Memetic Vector Quantizer are as follows

Step1: MA encodes the entire code book in a chromosome so that each meme encodes one code vector. Each code vector is assigned an index. An initial population of P individuals (chromosomes) is generated. If X be a set of training vectors of size M, $X=\{x_1, x_2 \ldots x_M\}$, then basic operation is classification of training vectors into N units. A code word (meme) is the centroid of the training vectors whose indices belong to a certain unit. Hence, each basic individual consists of N code vectors or code words (memes).

Step2: The local search algorithm is then applied to each individual in the initial population to improve the fitness of individual. The one step LBG algorithm is used as local search in the proposed algorithm. That is, each training vector is assigned to its closest meme (code vector) encoded in chromosome and then each code vector is updated as centroid of the training vectors whose indices belong to it.

Step3: The fitness function is used to evaluate the performance of each chromosome for the environment. The fitness value 'F' is computed as inverse of MSE, where MSE is Mean Square Error (MSE) between code book vectors in a chromosome and the given training vectors The fitness function is computed for each chromosome in the population.

Step4: The parent chromosomes for reproduction from the current population with a probability proportional to their fitness are selected. Selection is one of the key operators on MAs that ensures the survival of the fittest.

Step5: MA has a set of crossover and mutation operators that facilitates the exchange of information between two chromosomes on one hand and allows variation to be introduced to avoid trapping at local optima on the other. The crossover operator has always been regarded as a fundamental search operator in GAs since it exploits information about the search space that is currently available in the population. A number of cross over operators exist in the literature [2]. In the proposed algorithm, one point cross over operator is applied to the selected parent chromosomes producing two off springs Newly created off springs are added to the offspring set.

Step 6: Perform Gaussian Mutation to each member of the child population, with a very low probability. It enables new features to be introduced into a population. It also protects the individual s against irrecoverable loss of good features.

Step 7: Apply one step LBG algorithm to each chromosome of the offspring population to improve its fitness

Step 8: Replace the least fit chromosomes in the existing parent population by the newly generated offspring Chromosomes.

Step 9: Repeat step 4 to step 8 until the stopping criteria are met.

The algorithm terminates when the limit on the number of generations is exceeded or if there is no improvement in the fitness function of the best individual for five generations.

IV. EXPERIMENTAL RESULTS

To evaluate the efficiency of the proposed algorithm, MVQ has been implemented in Matlab 7.0 and run on Pentium IV computer, along with GVQ and LBG. The Lena image of resolution 256 x 256 is used to generate 4096 training vectors of dimension 16 (4 × 4). The generic code book consisting of k code vectors with dimension 16 is generated. A population size of 6 is used, that is 6 code books consisting of k code vectors of dimension 16 are generated initially. A crossover probability of 1 is used (i.e.) the two parents are always crossed in generating the new individuals. Uniform crossover operator is used.

Mutation probability of 0.01 is used. The algorithm terminates when the number of generations exceeds 100 or the fitness of the best chromosome does not improve for five generations. The fittest chromosome in the final population is selected as the generic codebook for vector quantization. Further, lossless Huffman encoding is applied to the indices generated by the encoder. Codeword assignment for the indices is based on the frequency distribution of the code vectors in the encoded image, more compression is achieved

Figure 2 compares the improvement of the fitness of the best chromosome for 100 generations while evolving the code book of different sizes using Memetic algorithm with that of Genetic algorithm.

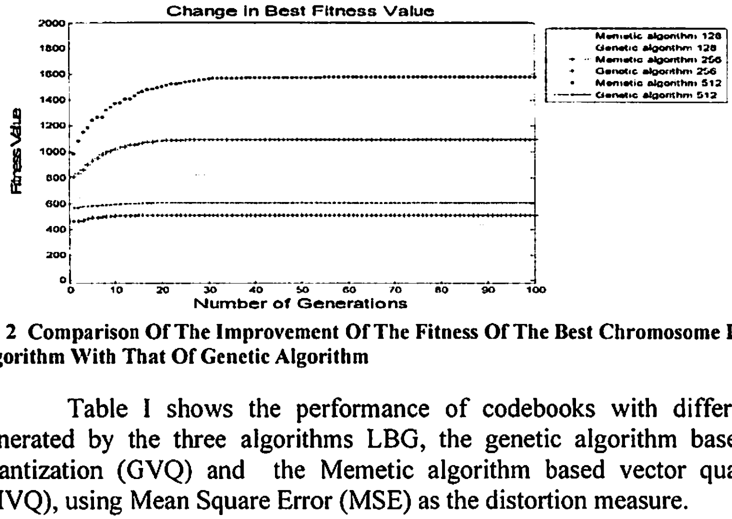


Fig 2 Comparison Of The Improvement Of The Fitness Of The Best Chromosome In Memetic Algorithm With That Of Genetic Algorithm

Table I shows the performance of codebooks with different sizes generated by the three algorithms LBG, the genetic algorithm based vector quantization (GVQ) and the Memetic algorithm based vector quantization (MVQ), using Mean Square Error (MSE) as the distortion measure.

Table I Performance Comparison Of LBG Algorithm, Genetic VQ And Memetic VQ In Terms Of MSE

	Code book size =128			Code book size =256			Code book size =512		
IMAGE	LBG	GVQ	MVQ	LBG	GVQ	MVQ	LBG	GVQ	MVQ
Cameraman	79.00	30.72	29.31	56.67	27.72	27.41	33.76	28.54	25.84
Aerial View	89.63	77.37	76.62	77.57	75.19	74.05	71.86	72.09	71.05
Lena	57.29	31.62	23.79	26.64	28.85	19.49	21.01	26.56	15.55
Glass	56.72	24.08	21.17	36.94	17.05	17.23	15.32	12.88	9.60
Fruit	81.57	29.64	26.96	39.05	24.07	24.22	27.36	22.45	20.69
Rice	75.30	40.57	37.87	41.50	38.03	34.81	33.60	35.09	33.98
Coin	103.92	29.84	29.73	32.94	27.68	25.36	25.66	24.60	22.79
Bird	60.26	16.38	13.68	18.22	13.22	12.06	13.31	12.53	10.08
Mosaic	93.41	31.29	26.46	49.79	20.84	20.21	19.56	16.99	14.03
Peppers	66.46	33.23	32.57	40.22	30.07	29.46	31.46	27.16	27.20
Moon	30.27	18.98	8.58	27.43	16.96	17.27	13.55	13.00	13.90
Eye	64.94	32.73	30.32	34.48	27.71	28.20	26.07	25.11	23.89
Blood	56.01	36.86	33.59	37.42	34.00	31.12	31.56	32.09	28.76
Mona	101.56	38.83	37.67	72.12	29.13	33.73	43.44	25.01	24.26

As the code book size increases the search space increases exponentially in VQ. Therefore for larger codebook sizes, the locally convergent algorithms could find very bad solutions. This is observed in our simulation studies. As can be seen from this table, the GVQ performs better than LBG algorithm in finding minimum. From the results reported for "Lena" image of code book sizes 256 & 512, it is found that GVQ may not always find the global optimum. It is also seen in the results reported for "Aerial View" for code book size 256. It is also shown that the MVQ proves to be effective in finding the global optimal minimum.

To show the efficiency of the proposed algorithm the objective quality of encoded images are measured using PSNR (Peak value Signal-to-Noise Ratio), which is defined as

$$PSNR = 10\log_{10} \frac{255^2}{\frac{1}{MN}\sum_{i=1,j=1}^{M,N}(x_{ij} - x'_{ij})^2}$$

Where x_{ij} is the value of the ij^{th} pixel in the original image and x'_{ij} is that of the reconstructed image.

Table II compares the three algorithms LBG, GVQ and MVQ in PSNR for different codebook sizes (128,256,512) with fourteen images. For different codebook sizes, the codebooks obtained by MVQ outperform those by LBG by 0.05 - 6.44 dB in PSNR and the codebooks obtained by MVQ outperform those by GVQ by 0.04 - 2.33 dB in PSNR. Thus the effectiveness of MVQ is proven. As can be seen from this table for the codebook size of 256, the percentage improvement of MVQ over LBG is 21.23 and that of MVQ over GVQ is 6.86.

Table II Performance Comparison Of LBG Algorithm, Genetic VQ And Memetic VQ In Terms Of PSNR IN dBs

	Code book size =128			Code book size =256			Code book size =512		
IMAGE	LBG	GVQ	MVQ	LBG	GVQ	MVQ	LBG	GVQ	MVQ
Cameraman	29.16	33.26	33.46	30.60	33.70	33.75	32.85	33.58	34.01
Aerial View	28.61	29.25	29.29	29.23	29.37	29.44	29.57	29.55	29.62
Lena	30.55	33.13	34.37	33.88	33.53	35.23	34.91	33.89	36.22
Glass	30.59	34.32	34.87	32.46	35.81	35.77	36.28	37.03	38.31
Fruit	29.02	33.41	33.82	32.21	34.32	34.29	33.76	34.62	34.97
Rice	29.36	32.05	32.35	31.95	32.33	32.71	32.87	32.68	32.82
Coin	27.96	33.38	33.40	32.95	33.71	34.09	34.04	34.22	34.55
Bird	30.33	35.99	36.77	35.52	36.92	37.32	36.89	37.15	38.10
Mosaic	28.43	33.18	33.90	31.16	34.94	35.08	35.22	35.83	36.66
Peppers	29.91	32.92	33.00	32.09	33.35	33.44	33.15	33.79	33.79
Moon	33.32	35.35	35.44	33.75	35.84	35.76	36.81	36.99	36.70
Eye	30.01	32.98	33.31	32.76	33.70	33.63	33.97	34.13	34.35
Blood	30.65	32.47	32.87	32.40	32.82	33.20	33.14	33.07	33.54
Mona	28.06	32.24	32.37	29.55	33.49	32.85	31.75	34.15	34.28

Table III compares the compression ratio(CR) achieved using LBG algorithm and MVQ for the fourteen standard images.

Table III Performance Comparison Of LBG Algorithm, And Memetic VQ In Terms Of Compression Ratio(CR).

IMAGE	Code book 128		Code book 256		Code book512	
	LBG	MVQ	LBG	MVQ	LBG	MVQ
Cameraman	2.26	4.40	4.65	5.07	6.23	6.78
Aerial View	2.36	5.46	5.52	6.65	7.47	8.12
Lena	2.29	5.03	5.26	6.14	6.98	7.62
Glass	2.64	4.28	4.26	4.82	6.37	6.35
Fruit	2.31	4.75	4.51	5.72	6.46	7.22
Rice	2.17	4.37	4.91	5.41	6.57	6.70
Coin	1.75	3.59	3.40	4.38	4.99	5.46
Bird	2.05	4.11	4.51	5.03	5.95	6.48
Mosaic	2.32	3.99	4.23	4.81	5.85	6.18
Peppers	2.37	4.93	5.05	5.90	6.86	7.47
Moon	1.92	3.04	2.82	3.15	7.00	4.26
Eye	2.06	4.13	4.43	5.14	6.12	9.40
Blood	1.86	4.27	4.16	5.27	5.82	6.73
Mona	2.38	4.52	4.18	5.32	6.09	7.19

It is evident from the Table III that the image compressed by the MVQ is quite superior to the LBG algorithm.

Figs. 3, 4, 5 and 6 shows the variation of CR and PSNR with respect to the variation of the size of the code book for the standard images Cameraman ,Lena ,Bird and Eye. From these figures it is inferred that the MVQ algorithm provides high PSNR and Compression Ratio (CR) than the conventional LBG algorithm for different codebook sizes. This indicates that MVQ provides better visual quality and competitive compression ratio than other algorithms.

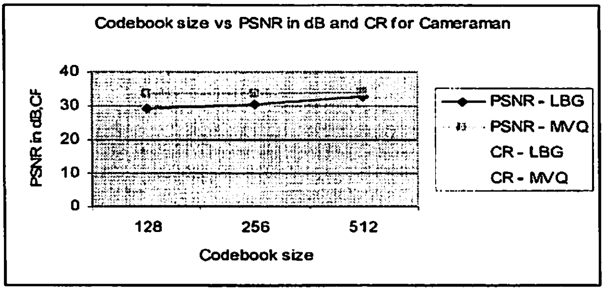


Fig 3 Codebook size vs PSNR in dB and CR for Cameraman

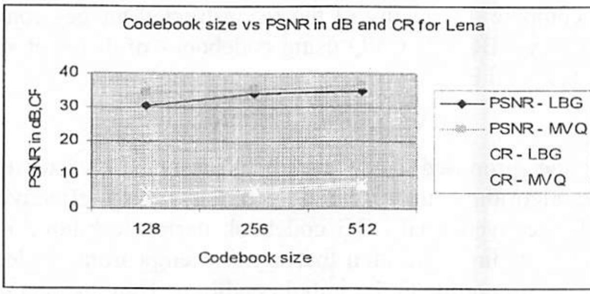


Fig 4 Codebook size vs PSNR in dB and CR for Lena

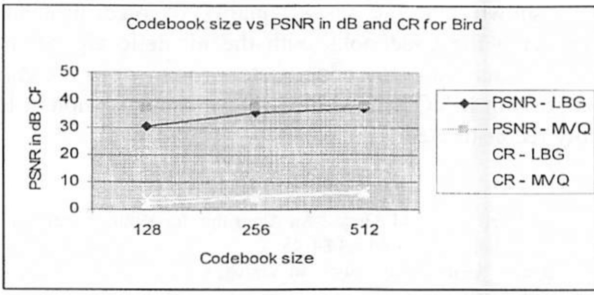


Fig 5 Codebook size vs PSNR in dB and CR for Bird

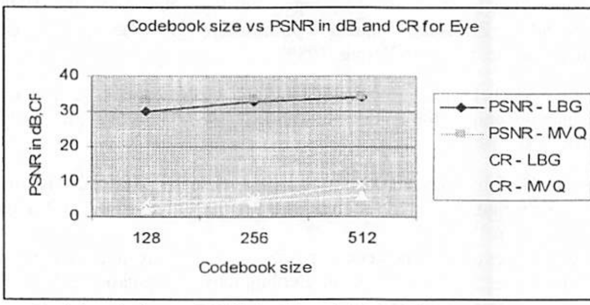


Fig 6 Codebook size vs PSNR in dB and CR for Eye

Fig 7 shows the Original images Cameraman and Bird.

(a) Cameraman

(b) Bird

Fig 7 Original Images

Figs. 8 and 9 compare the quality of the reconstructed images compressed by MVQ with those by LBG and GVQ using codebooks of different sizes for the images Cameraman and Bird.

V. CONCLUSIONS

This paper proposed a new codebook design for vector quantization based memetic algorithms that have proved to be very effective in image compression. The conventional LBG codebook design technique is a kind of steepest descent searching algorithm that is not exempt from the local minima problem and very susceptible to the initial condition. These results demonstrate that the search performance obtained by MAs is better than that obtained by the GA alone. It is shown that the visual quality of reconstructed images is improved by evolving the codebooks with the memetic algorithm. It can be further improved by increasing the code vectors (memes) in the chromosomes. The proposed algorithm MVQ as a whole achieves compression at low bit rates with good quality reconstructed

REFERENCES

[1] Y. Linde, A. Buzo and R .M. Gray. "An Algorithm for Vector Quantizer Design" IEEE Trans.Commun. Jan.1980, COM-28:84-95

[2] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, 1989.

[3] L. Davis, Handbook of Genetic Algorithms.Van Nostrand Reinhold, New York, 1991

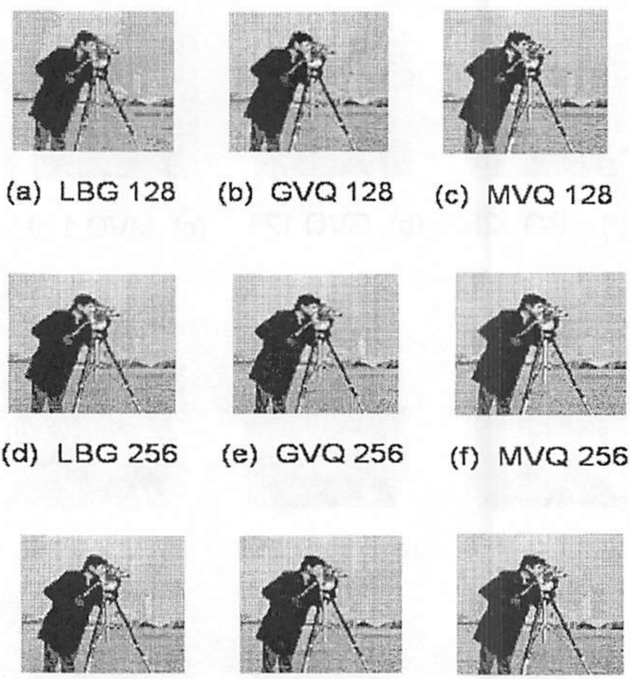
[4] D.Saloman, "Data Compression ", Springer- Verilag New York, Inc. 1998

[5] A. Torn and A. Zilinskas, "Global Optimization", Volume 350 of Lecture Notes in Computer Science, Springer-Verlag, 1989.

[6] J. A. Miller,W.D. Potter, R. V. Gandham, and C. N. Lapena, "An evaluation of local improvement operators for genetic algorithms," IEEE Trans.Syst., Man, Cybern., vol. 23, pp. 1340–1351,. 1993.

[7] N. Shahidi, H. Esmaeilzadeh, M. Abdollahi, C. Lucas, "Memetic Algorithm based Path Planning for a mobile robot," International Journal of Information Technology, Vol. 1, Num. 4, 2004, pp 174-177.

[8] Smith, J.E,"Co-evolving memetic algorithms: a learning approach to robust scalable optimization"The 2003 congress on Evolutionary computation,2003 CEC'03"Volume 1, 8-12 Dec. 2003,pp 498-505



(a) LBG 128 (b) GVQ 128 (c) MVQ 128

(d) LBG 256 (e) GVQ 256 (f) MVQ 256

(g) LBG 512 (h) GVQ 512 (i) MVQ 512

Fig 8 Results On Cameraman Image

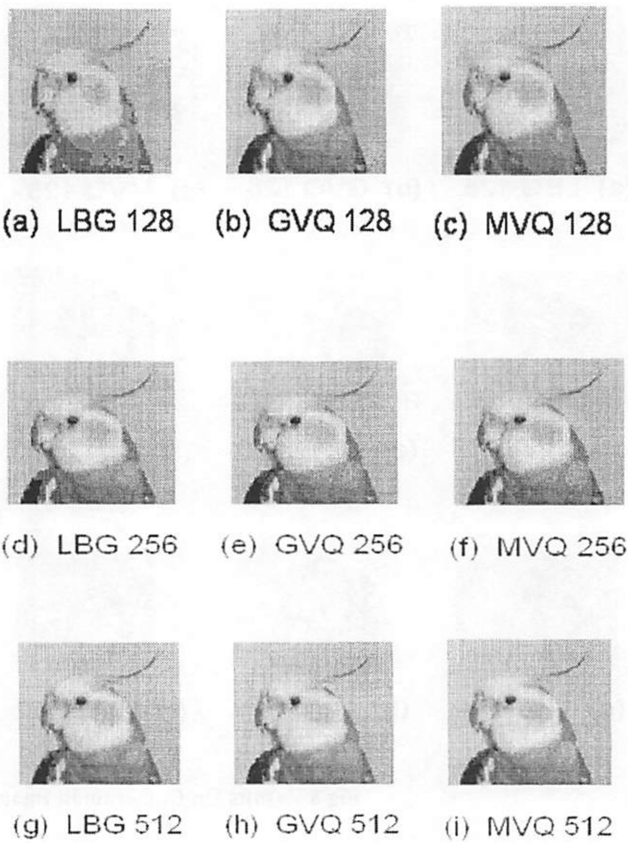


Fig 9 Results On Bird Image