Discrete wavelet transformation of an image based on genetic-algorithm clustering

Dr. K.Vivekanandan [1], P.Krishnakumari [2]
[1] *School of management, Bharathiar University, Coimbatore, Tamilnadu, India*
[2] *Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women , Coimbatore -641044 , Tamilnadu, India*
E-mail: kkjagadeesh@yahoo.com

Abstract: In this paper the image compression problem is analyzed using genetic clustering algorithms based on the pixels of the image. The main problem to solve is to find an algorithm that performs the clustering efficiently. The possibility of solving clustering problems with genetic algorithms provides optimal solution. The present work makes use of genetic clustering algorithms to obtain an ordered representation of the image and then applies the DWT (Discrete Wavelet Transformation) to compress the image. The image quality is measured using PSNR measure. The genetic algorithm (GA)-based compression image has more PSNR value compared to the image without GA. It shows that the quality of the image is good for GA based compression. Also it reduces the problem of memory utilization efficiently due to compression and at the same time quality of the image is maintained due to GA.

Keywords: Genetic algorithms, clustering, image compression, discrete wavelet transformation.

Introduction

Digital images and videos are still demanding in terms of storage space. For many years lossy image compression is used which makes use of only a subset of the original data to approximate the input image. Several problems exist with the techniques that are used to determine the best representatives of the data. Heuristic search methods, which attempt to perform a manual search of all data points to determine the closest ones, are costly in terms of processing time and memory. It is the case that these heuristic methods cannot be used for processing any significant amount of data, as the storage cost is far too high.

Ismail & Kamel (1989) proposed an algorithm that alternates between a depth first search and a breadth first search to minimize the objective function. In these algorithms for each iteration, objects are systematically moved to different clusters, whenever the action decreases the value of the objective function. The greedy nature of these algorithms can make them get stuck to a local minimum. Only a few techniques exist to provide an optimum solution in terms of quality and time for imaging applications. From a theoretical standpoint, the Genetic Algorithm (GA) is deemed to be the most suitable (Goldberg, 1989; Holland, 1975). Chiu and Liu (1996) state that compared to the available search methods- calculus, enumerative and random, the GA is the most robust. Various factors make the GA a strong candidate. Each time the GA is exposed to a new set of training data, its solution is further optimized.

Klein and Dubes (1989) have applied simulated annealing, but the main disadvantages of this method are the great amount of execution time and that an efficient schedule for a simulated annealing algorithm is very difficult to achieve. Bhuyan (1991) considers the problem of partitioning N objects in M disjoint clusters using genetic algorithms to obtain a suitable object permutation. The execution of GA is relatively controllable and flexible to cater for a specific application (Brown *et al.*, 1989). Hwang and Hong (1999) remarked also on the insensitivity of the Genetic Algorithm to initial starting conditions, allowing flexibility in domains where data can vary considerably. Although genetic algorithms are regarded as one of the strongest candidates for optimal search, few attempts were made to exploit those robust features embedded in GAs in image processing, in which many problems can be refreshed in the framework of optimization. This paper serves such an attempt to explore new directions of developing alternative techniques in image compression.

A Brief introduction to Genetic Algorithms

Genetic algorithms are search algorithms based on natural genetic and selection combining the concept of survival of the fittest with a structured interchange. These concepts involve the preservation of the characteristics of the best exponents of the existing ones in the next generation. Moreover, there is a possibility of introducing aleatory changes in the newer generation composition by means of cross over and mutation operations. This aleatory component prevents getting stuck into a local maximum and reaches global maximum. This would represent one of the main advantages of genetic algorithm in opposition to the traditional search methods like the gradient method. Another advantage is its utility for real time applications. In spite of not providing the optimal solution to the problem, it provides even better solution to a complex problems within a shorter time compared to that of traditional methods.

Following a major lines of research in the area of evolutionary computing (Chen *et al.*, 1999; Chiu & Liu,1996; Hwang & Hong, 1999; Sexton & Gupta, 2000), the proposed genetic approach bears the most representative structure and contains all major components for a genetic algorithm (Goldberg, 1989; Holland, 1975). The input image data is sorted in structures that allow GA operators to work on it successfully. The Genetic Algorithm works on the data set to obtain the optimum representative blocks. The original image is transformed using the derived representatives, into a compressed file.

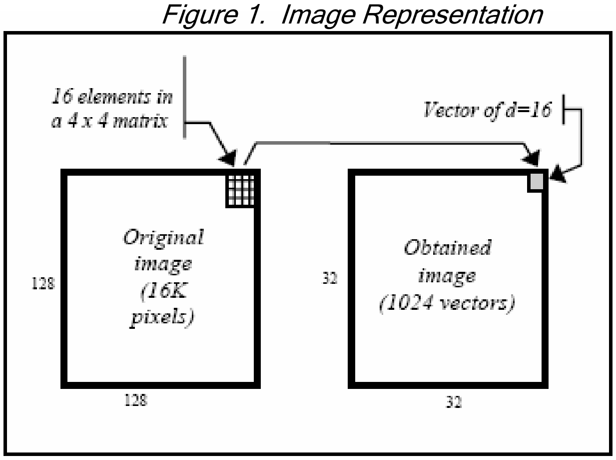
Clustering with Genetic Algorithms

Clustering purpose is to divide a given group of objects into a number of clusters, in order that the objects in a particular cluster are similar among the objects of the other ones. This technique tries to distribute N object in M clusters according to the minimization of some optimization criterion (Aldenferder & Blashfield,1984; Gordon & Henderson, 1977). Once the optimization criterion is selected, the clustering problem provides an efficient algorithm in order to search the space of the all possible classifications and to find one on which the optimization function is minimized. The problem is to classify a group of samples. These samples form clusters of points in n-dimensional space. These clusters form groups of similar samples. The more formal procedures use an optimization criterion such as minimizing the distance of each sample to the clusters centre, which can be considered as the best optimization of a cluster. This means a unique point X better represents all points from this cluster. This optimization criterion was used in the proposed work and the minimization process is performed by genetic algorithm.

Clustering technique applications to the image compression

The subject image is first divided up into blocks of pixels. The size of the blocks is important as it affects both quality and compression. Large blocks increase compression ratio at a cost to quality whereas the reverse is true for small block size. To achieve an appropriate balance, 4 x 4 image blocks is chosen for the design of the proposed algorithm. The image of size 128 x 128 pixels is considered, each one can take a value between 0 and 255, thereby giving a resolution of 256 grey tones. In this way, a byte is enough to code each pixel; therefore the size of the image is 17170 including header information. Reading the image file is performed constructing arrays of 4x4 elements as it is shown in the following picture.

Figure 1. Image Representation



The figure 1 shows that the final image has 32 x 32 =1024 vectors of 16 bytes each .The clustering idea can be implemented to these vectors that contain information of the image. As the amount of possible grey tones with this representation scheme is 256, the same can be applied to 1024 vectors. Starting from the array of 32 x 32 vectors of 16 elements each, 8 groups of 8 x 16 vectors each are generated resulting in 128 vectors representing a chromosome. They are formed by dividing the original array in 8 rectangles.

Once the 8 groups are formed, the clustering algorithm is executed to carry out the classification by parts. Each of these groups is independently classified into 32 clusters. Once the clustering for the 8 groups is finished, 256 clusters will be obtained. Each chromosome is a vector of 128 elements (bytes).

The genetic optimization can be regarded as a process of obtaining the true group representative by means of generating a block that bears the closest resemblance to all members inside the group. The genetic algorithm uses a traditional layout for its operation, which includes fitness, selection, crossover, mutation and stopping criteria. The system uses a population of genes upon which the fitness, selection, crossover and mutation would operate. These are set as a number of 4x4 blocks, (here 128), an arbitrary figure that achieves a reasonable balance between processing cost and search speed.

Fitness evaluation

The optimization criterion that has been used is the Square of the Sum of Errors. As the aim is to minimize the distance of each vector to the center of the cluster to which it belongs to, the fitness function to maximize:

$$f = \begin{cases} M - e & \text{si } M - e \geq 0 \\ 0 & \text{si } M - e < 0 \end{cases}$$

where

M is a constant that in the first place is equal to the maximum error possible and e is given by:

$$e = \sum_{i=1}^{NC} \sum_{j=1}^{n^i} \left\| \overline{x}_j^i - \overline{z}^i \right\|^2$$

Where:

NC: number of clusters

n^i: number of vectors in cluster i

$\overline{x}_j^i$: vector j of cluster i

$\overline{z}^i$: mean value of cluster i

The mean value of the cluster i is calculated as:

$$\overline{z}^i = \frac{\sum_{i=1}^{n^j} \overline{x}_i^j}{n^j}$$

The problem encountered here was that due to the great value of the maximum constant computed for the fitness function, all the fitness values for the chromosomes of a population were too close each other. In order to solve this problem, the fitness function can be scaled in the following way:

$$f' = a \cdot f + b;$$

a and b were calculated considering the fitness average to be the same for the original function *f(x)* and for the function transformed *f'(x)* using the following conditions:

$$a \cdot f_{min} + b = 0; \qquad a \cdot f_{prom} + b = f_{prom}$$

Solving these equations a and b are as follows:

$$a = \frac{f_{prom}}{f_{prom} - f_{min}}$$

$$b = \frac{f_{prom} \cdot f_{min}}{f_{min} - f_{prom}}$$

Crossover

The two strongest parents returned by the fitness function are used to populate the 128-gene population. Normally, single point crossover is used in such problems; Hwang and Hong (1999) used such a crossover in their vector quantiser. Single point crossovers simply split the parent strings in two and join the halves from each parent together. This length at which this split occurs can be either fixed or chosen at random. Here single point crossover with a crossover probability of 0.9 is chosen.

Mutation

Many genetic algorithms employ a mutation operator that randomly reverses bits in a binary string. For imaging applications, mutation can be designed by implementing a 'smoothing' mutation operator, similar to the method used by Chen *et al.* (1999). When mutation occurs decided by its relevant probability, a target pixel is further chosen at random. The mutation operates by setting the intensity of the target pixel to the average intensity of its surrounding pixels (bar edges). This results in the subtle intensity changes that occur frequently in images, which are effectively invisible to the human eye. Mutation rate used in the proposed algorithm is 0.1. Total number of generations used for the termination criteria is 100. The proposed GA quantization is integrated with DWT for image compression. The quality of the image is measured using PSNR.

Discrete wavelet transform (DWT)

In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. Wavelet compression is a form of data compression well suited for image compression (sometimes also video compression and audio compression). The goal is to store image data in as little space as possible in a file. Wavelet compression can be perfect, lossless data compression, or a certain loss of quality is accepted with lossy data compression.Following the major lines of research using wavelets (Arrival *et al.*,1993; Conforto *et al.* 1995; Faloutsos et al.,1994; Hsieh & Tsai, 1996; Karayiannis & Pai, 1996; Karayiannis & Pai, 1995; Kin Pong Chan & Ada Wai-Chee Fu, 1999; Lo *et al.* 1996; Mohsenian *et al.*1993;Moore 2005;Michael Peterson 2007), the proposed algorithm uses DWT for compression based on (Burrus *et al.*,1998).

PSNR

The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. The more per pixel difference between the pictures is the less is PSNR value.

Results

The proposed Genetic algorithm compression is implemented using MATLAB for a set of sample images. The image is compressed in a lossy manner by applying wavelet transform to an entire image. Peak signal –to- noise ratio (PSNR) is used as an objective measure of image quality. GA clustering considers the directions of minor variations of the pixels tones, avoiding sudden changes and thus produces optimal representation of pixels for transformation.

The following output shows a sample image compressed based on GA clustering using Discrete Wavelet Transformation:

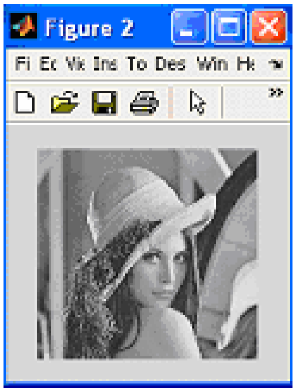


Fig 2. Original figure; Fig.3 GA based DWT Compression

To estimate the performance of the proposed genetic clustering algorithm, sample images with corresponding PSNR measure were shown in table 1.

Table 1. Performance of the proposed algorithm

	PSNR (dB)	
	The Proposed GA based DWT	DWT
Image 1	20.20	19.83
Image2	22.21	21.55
Image 3	21.20	20.04
Image 4	19.88	18.45
Image 5	19.03	18.05

The proposed GA based compression image has more PSNR value compared to the image without GA and this shows that the quality of the image is good for GA based compression.

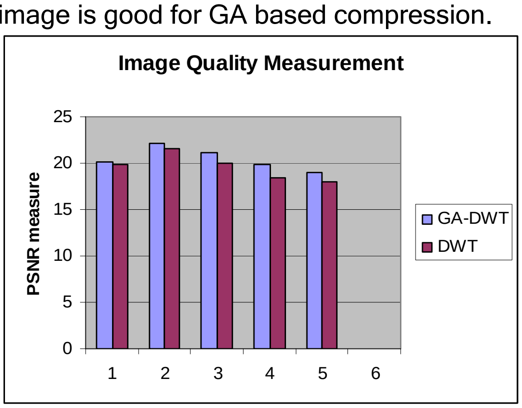


Fig 4. Image quality measure

The results show that the GA outperforms the DWT in terms of both PSNR and visual quality. An important aspect of the proposed genetic approach is represented by the design of fitness function, crossover and mutation. The mechanisms it employs to achieve optimal performance on multi dimensional data are useful for any optimization problem where complicated data elements are used.

Conclusion

In this paper, the technique of clustering with genetic algorithm was applied to image compression. Basic operators of selection, crossing over and mutation were utilized. The main idea is to divide the problem into parts and to apply the clustering technique to each part independently and this produces optimal representation for transformation. The results show that the GA outperforms the DWT in terms of both PSNR and visual quality. Further research along this line is the integration of Genetic Algorithm quantization with other transform such as DCT. The quantization of these signal coefficients, if successful, will yield high compression and high PSNR values.

References

1. Arrival R, Faloutsos C, and Swami A (1993) Efficient Similarity Search in Sequence Databases. In: Proceedings of the 4th Int'l Conference on Foundations of Data Organization and Algorithms, Chicago, pp: 69-84.
2. Aldenferder M and Blashfield R (1984) Cluster Analysis, Sage Publications, Beverly Hills.
3. Bhuyan J (1991) Genetic algorithm for clustering with an ordered representation.In: Proceedings of Fourth International Conference on Genetic Algorithms, pp: 408-415.
4. Brown D, Huntley R and Spillane C (1989) A parallel genetic heuristic for the quadratic assignment problem. In: Proceedings of third international Conference on Genetic Algorithms, pp: 406-415.
5. Chen Y, Nako Z and Arakaki K (1999) Restoration of grey images based on a genetic algorithm with laplacian constraint. In: Fuzzy sets and systems, 103.
6. Chiu C and Liu P (1996) Image reconstruction of a perfectly conducting cylinder by the genetic algorithm. *IEEE Proc.-Microw. Antennas Propag*. 143 (3), 40-46.
7. Conforto S. *et al*. (1995) High-quality compression of echo graphic images by neural networks and vector quantization. *Med. Biol. Eng. Computations*. 33 (5), 695- 698.
8. Faloutsos C, Ranganathan M, and Manolopoulos Y, (1994) Fast subsequence matching in time-series databases. In: Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, Minneapolis, pp: 419-429.

9. Goldberg DE (1989)Genetic algorithms in search, optimization and machine Learning, Addison –Wesley.
10. Gordon A and Henderson J (1977*)* An algorithm for Euclidean sum of squared Classifications. *Biometric*. 33, 355- 362.
11. Holland JH (1975) Adaptation in Natural and artificial systems, Univ. of Michigan Press, Ann Arbor, Mich.
12. Hsieh CH and Tsai JC (1996) Lossless compression of VQ index with search-order coding. *IEEE Trans. Image Process*. 5 (11), 1579-1582.
13. Hwang W and Hong S (1999) Genetic entropy-constrained vector quantiser design Algorithm. *Opt. Eng*.38, 12-19
14. Ismail M and Kamel M (1989) Multidimensional data clustering utilizing hybrid search strategies. *Pattern Recognition*. 22 (1) 75-89.
15. Karayiannis NG and Pai PI (1996) Fuzzy algorithms for learning vector Quantization. IEEE Trans. *Neural Networks*. 7 (5), 1196-1211.
16. Karayiannis NG and Pai PI (1995) Fuzzy vector quantization algorithms and their application in image compression. *IEEE Trans. Image Process*. 4 (9), 1193-1201.
17. Klein R and Dubes D (1989) Experiments in projection and clustering by simulated annealing. *Pattern Recognition*. 22, 213-220.
18. Kin Pong Chan and Ada Wai-Chee Fu (1999) Efficient time series matching by Wavelets. In: Proceedings of the 15th International Conference on Data Engineering 23-26 March 1999, IEEE Computer Society, Sydney, Australia, pp: 126-133.
19. Lo SB *et al*. (1996) On optimization of orthonormal wavelet decomposition: implication of data accuracy, feature preservation, and compression. *SPIE Proc*. 27, 201-214.
20. Mohsenian.N, Rizvi SA and Nasrabadi NM (1993) Predictive vector quantization using a neural network approach. *Opt. Eng*. 32 (7), 1503-1513.
21. Sexton RS and Gupta JND (2000) Comparative evaluation of genetic algorithm and back propagation for training neural networks. *Information Sciences*. 129, 45-49.
22. Sheikh HR, Bovik AC, Cormack L, and Wang Z (2003) LIVE image quality assessment database. http://live.ece.utexas.edu/research/quality.
23. Burrus SC, Gopinath RA, and Haitao Guo (1998) Introduction to Wavelets and Wavelet Transforms, A Primer, Prentice-Hall, Inc.
24. Moore F (2005) Genetic algorithm for optimized reconstruction of quantized signals, *IEEE congress on evolutionary computation* ,1,105-111
25. Michael Peterson (2007) A satellite image set for the evolution of image transforms for defense applications , *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, United Kingdom, 2137 - 2144