مجلة جامعة بابل / العلوم الصرفة والتطبيقية / العدد (4) / المجلد (17) : 2009

Using Genetic Algorithm to Generate Optimal Codebook for Image Compression

Suhad A. ALI
Science collage for Women Babylon university

Abstract

In this paper we proposed new method for image compression using vector quantizer. We use genetic algorithm to generate optimal codebook form group of codebooks to represent the image. Simulation results are presented to show the superior performance of the proposed method in terms of peak signal-to-noise ratio as design.

الخلاصة

تم في هذا البحث اقتراح طريقه جديدة لضغط الصور باستخدام الخوارزمية الجينية لتوليد codebook التي بناءها لتمثيل الصورة . توضح النتائج أداء الخوارزمية المقترحة باستخدام مقياس نسبة الإشارة إلى الخطأ PSNR.

1.Introduction

Vector quantization is an effective technology for data compression and it has been successfully used in speech encoding and image compression[Salmon. D 2004]. Vector quantization is the process of mapping a vector that can have many values to the vector that has a smaller (quantized) number of values. An N-level k-dimensional vector quantizer is a mapping , $Q: R^k \to C$ that assigns to each training vector, $X = (x_1, x_2, x_3, ..., x_k)$, a reproduction vector, $Y_i = Q(X)$, drawn from a finite codebook, $C = (Y_1, Y_2, ..., Y_N \left| Y_i \in R^k \right.)$. Both the encoder and the decoder work using the same codebook. In VQ system the encoder determine the closest codeword in the Codebook against an incoming vector and thereby mapping the input vector into a set of index numbers. The decoder's function is only a table look up fetching out the code vectors corresponding to the received index number[Begum. *et al.,* 2005].many traditional search method can hardly find the best representative of data especially when the data become large. To determine the closest ones are costly in terms of processing time and memory. The first attempt by Linde, *et al.,* which called Generalized Lloyd Algorithm (GLA) or(LBG) uses a finite set of training sequences as the data source and employs an iterative refinement. Given an initial codebook, the algorithm computes the nearest locally optimum codebook only[Linde. *et al.,* 1980].

Po [Po. 1990] Attempt to reduce the computation time by reducing the dimension measure in the LBG algorithm, but the codebooks generated by this method is slightly degraded. Ma and Chan [Ma. and Chan, 1991] propose a maximum descent algorithm that compared with LBG algorithm, the codebook performance is improved and the computation time is reduced. However, this algorithm can only obtain a local optimal code bock. Torres and Huguet [Torres and Huguet. 1994] and Ra and Kim [Ra. and Kim,1993], They are makes some slight modification in the LBG search to reduce the computation time , however they cannot improve the codebook performance.

Genetic algorithms (GAS) [Goldberg,1989] are emerging as widely accepted optimization and search methods. These search methods are rooted in the mechanisms of evolution and natural genetics. They have a high probability of locating the globally optimal solution in a multimodal search space. The research that using genetic algorithms

with vector quantizer are : Armstrong and Jiang [Armstrong. and Jiang, 2000] used a simple genetic algorithm for image quantization to find near – optimal codeword. their simple GA produce the codevector that represent the all codevectors belong to the same index in the codebook, but their algorithm suffer of prematurely and may lack in local optimum during the search because selecting the best chromosomes at each generation for crossover with crossover probability very low and very low mutation probability That lead to remain the same two chromosomes the best to be selected again in the next Generation. describes a genetic algorithm for the problem of codebook design. Zheng, *et al.*, [Zheng. *et al.*, 1995] used The genetic algorithm's chromosomes represent partitions of the training set; each vector maps to the codeword that is the centroid of its set in the partition. To speed up its operation, the genetic algorithm uses fitness inheritance to assign fitness values to most new chromosomes, rather than evaluating them. Tests using five standard digitized images compare the genetic algorithm to a popular non-genetic algorithm for codebook design. The genetic algorithm is found to be effective, but slow. in this paper the proposed algorithm is to find optimal codebook for testing image using genetic algorithm.

2-Genetic Vector Quantization

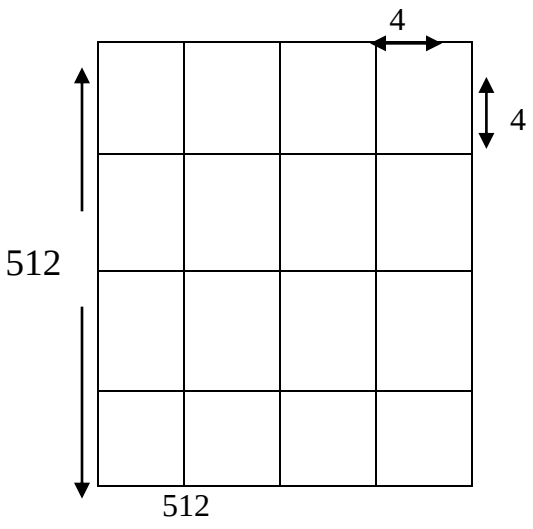
The aim of codebook design is to find the best classification of training vectors. Using the Darwinian ideas of the survival of the fittest and interbreeding of fit parents, the genetic algorithms can be used as an efficient searching mechanism. Genetic algorithm operate in a more stochastic manner than traditional search algorithm. This guided randomness allows the genetic algorithm to find optimal solutions much quicker than their deterministic counterparts. The proposed method is to find optimal codebook using genetic algorithms which uses a traditional layout for its operation, which includes fitness, selection, crossover, mutation and stopping criteria. Given the number of training vectors M and the number of code vectors N, the codebook design is to classify the M training vectors into N clusters, before the genetic algorithm is applied the subject image is divided up into blocks of pixels. The size of blocks is important as it effect both quality and compression. Large blocks increase compression ratio at accost quality whereas the reverse is true for small block size. To achieve an appropriate balance, we chose 4×4 image blocks for the design of proposed genetic image compression as shown in figure(1). This size provided reasonable compression with the additional advantage that no padding is needed when the training image are select of 512×512 in size.

3- Proposed Algorithm

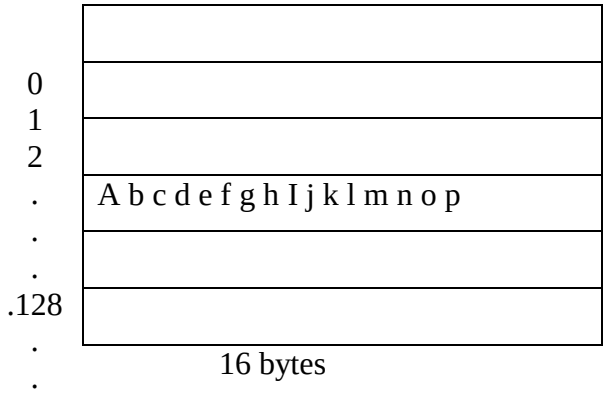
The following steps describe the propose algorithm:

3-1 **Genetic Initialization:-**

In this paper, each chromosome represent a codebook. The label I (I =1,2,3,……..,M) of the training vector is viewed as agene, thus the basic individual is made up of M genes which can split in to N units, each unit is made up of several labels which belong to this unit. N codewords is the center vector of the training vectors in each unit. When image is quantized, each 4×4 block only reference its codeword using 8-bits. The data structure of codebook will be an array of tow dimension the number of rows will be 256 this is represent the length of codeword, while the number of columns in the codebook will be varies according to the input image size.



a. original image 512×512 divided into 4×4 blocks.



B. codebook with 256 16-bytesentries.

a	b	c	d
e	f	g	h
i	j	K	L
M	N	O	p

c. a block image compressed with vector #122

Figure(1) : Quantizing with a codebook

The initial representative of population will be chose randomly as illustrate in the following code:

```
Algorithm Initialize Codebook
   For I ←1 to popsize
    For j←1 to N
     For k←1 to 16
      l=random(M)
      Pop[i].codebook[j,k]=image[l,k]
     Next k
   Next j
  Next i
End of Initialize Codebook
```

Then the blocks of image is classify to each unit in codebook using the squared error distortion using the following formula

$$d(X,Y_j)=\sum_{l=1}^{k}(x_l-y_{jl}).....(1)$$

Where X represent the block of training image

Y represent the unit j in the codebook.

Algorithm Initialize _generation

```
For i←1 to popsize
  Dist=0
 For s←1 to M
   For j←1 to N
    Dist=dist+min(function_RSE(block(s)-pop(i).codebook(j))
  Next j
 Next s
Next i
```

End Initialize _generation algorithm

Where M represent the number of training set, N number of units in codebook in our algorithm N=255.

The next step is calculation the AvgCodebook which represent the average of each unit in codebook according to the formula

$$\text{AvgCodebook}_r=\frac{\sum_{i=1}^{n_{lr}^{(t)}} x_{a_i^{(t)r}}}{n^{(t)_{lr}}} \quad \ldots\ldots\ldots (2)$$

Where $a_{li}^{(t)_r}\ (i=1,2,3,.....,n_{lr}^{(t)})$ ≡ gene I of unit r

r ≡ the number of units in the codebook (r = 1,2,..,N).

n^r ≡ the number of blocks in unit r

x a_i^r ≡ the vector whose label is a_i^r

and $n^1 + n^2 + \ldots + n^N = M$.

3-2 Evaluation of the individuals: each individual is evaluated by using fitness function. The chromosome that have high fitness value will be the best representative chromosome for study image. For each individual, the program calculate the overall distortion according to the following formula

$$D=\sum_{r=1}^{N}\sum_{i=1}^{n_{lr}^{(t)}}(X_{a_{li}^{(t)r}}-Y_r^{(t)})^2.....(3)$$

According to above formula each gene is compared to each unit in the codebook it is currently working on. Fitness function for each chromosome calculated according to the formula

$$fitness=1/D.....(4)$$

3-3 Selection: in this step, we select two parents for crossover and to produce new offspring(s).in the proposed algorithm I used Binary Tournament Selection (BIS), In this method, two different individuals are selected randomly. The individual that has higher

fitness will win to be added to the crossover mate. If the fitness value of the first individual is equal to the fitness value of the second individual, one of them will be selected randomly .

3-4 Crossover: The crossover property recognized the GA from other evolutionary algorithms (evolutionary strategy and evolutionary programming). Crossover is the process of exchanging the parents' genes to produce one or two offspring that carry inherent genes from both parents. I used Two point crossover (2x): It operates by picking two random points within two parent chromosomes, and then exchanging the genes between these points in each parent to produce two new offspring. This is performed with a certain probability, otherwise the two parents are copied as offspring's.

3-5 Mutation: in the proposed method, we select gene randomly from each individual and replace the center of that gene with mean to make this pixel take value nearest to its neighbor pixels .

3-6 Termination criteria :

In the proposed algorithm we used number of generation and error criteria as stopping criteria in this method. When one of the two condition is satisfied the program will stop.

4-Experimental Result:

In this section, we present computer simulation results to evaluate the performance of the proposed codebook design algorithm. The simulated results can having using gray scale images 512×512 . Performance is quantified by peak signal-to-noise ratio (PSNR) and which are defined as

$$PSNR = 10 \log_{10}\left(\frac{255^2}{MSE}\right) \text{.....(5)}$$

The probability of mutation is pm=0.001, and probability of crossover is pc=0.05.

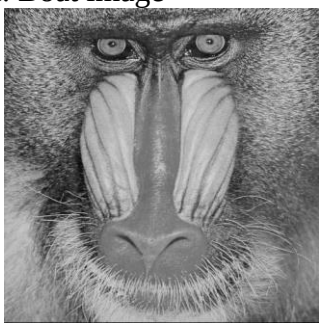
the following images are used as test image in program


a. Lena image


b. Zelda image


c. Boat image


d. Couple image


e. Baboon image


f. House image

figure(2) Test Images

Table (1)presents the performance of the proposed method in terms of PSNR

Table(1) Performance of the proposed method on test images

Images	PSNR (db)
Lena	31.50322
Zelda	33.6522
Boat	28.30809
Couple	27.8777
Baboon	23.6129
House	34.9086

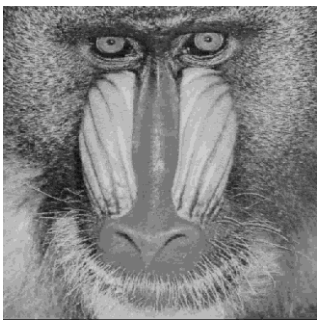
the result of reconstructed images of table (1)is shown in the following figure.

a. Reconstruct Lena image

b. Reconstruct Zelda image

c. Reconstruct Boat image

d. Reconstruct Couple image



e. Reconstruct Baboon image

f. Reconstruct House image

figure(3) Reconstructed Image

5- Conclusions

In this paper, we proposed a genetic image compressor to chose optimal codebook that represent the study image. The result show that GA give best result in term of PSNR. Because the object of the proposed algorithms is design optimal codebook, and each chromosome will represent codebook then there is some late while running program.

References

Armstrong. A and Jiang. J, (2000). Genetic algorithms for image compression.

Begum. M, Nahar.N, K. Fatima, M.K. Hasan, and M.A. Rahman, (2005). An efficient algorithm for codebook design in transform vector quantization.

Goldberg. D.E, (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Massachusetts.

Linde.Y, Buzo. A and Gary. R.A, (1980). An algorithm for vector quantizer Design, IEEE Trans. Commun., vol. 28(1), pp. 84-95, January .

Ma. C.K and Chan. C.K., (1991). Maximum descent method for image vector quantization , Electronic Letters, vol. 27, No. 12, pp. 1772-1773.

Po. L.M, (1990). Novel subspace distortion measurement for efficient implementation of image vector quantization, electronics Letters", vol.26, No. 29, pp. 480-482.

Ra. S.W and Kim. J.K, (1993). A fast mean-distance ordered partial codebook search algorithm for image vector quantization, IEEE Trans. Circuits syst. II, vol. 40, pp. 576-579.

Salmon. D, (2004). Data Compression: The complete reference, 3rd edition, Springer-Verlag new York, Inc.

Torres. L and Huguet. J, (1994). An improvement on codebook search for vector quantization, IEEE Trans. Communication, com-42, pp. 208-210.

Zheng. Y; Julstrom, B.A.; Cheng. W, (1999). Design of vector quantization code books using a genetic algorithm, Evolutionary Computation, IEEE International Conference, vol. 13,pp. 525-529.