CODEBOOK DESIGN FOR VECTOR QUANTIZATION USING GENETIC ALGORITHM

Wei-Hua Andrew Wang, Shih-Chun Yang and Chun-Liang Tung *
Department of Industrial Engineering and Enterprise Information
Tunghai University
Taichung (407), Taiwan

ABSTRACT

Genetic algorithms have been widely used to solve optimization in many fields such as multi-objective optimization, Fuzzy Optimization, and scheduling problem. Vector quantization, a basic method, is adopted by image compression technology and has a better performance than scalar quantization. Hence, it is worth to study how to apply genetic algorithms on the optimal design of codebook generation in vector quantization, where a codebook could minimize the average distortion between a given training set and the codebook. The advantage of the traditional genetic algorithms will be used in this paper, different from LBG training method, to evolve out a better codebook which is the nearest representative one by a fitness function in vector quantization. We take the random combinations of codebooks of the training samples as the initial population. Peak Signal-to-Noise Ratio is used as the fitness value. Using Two-point crossover and mutation process will get a better codebook finally after the evolution iterations in the proposed paper. Proved by our experiment, the proposed method, Simply Genetic Codebook Algorithm, can evolve and generate a better codebook through the whole experiment. To generate and acquire the high-quality codebook, the efficiency of codebook generation will be considered improved by imbedding stochastic model in the future.

Keywords: Genetic Algorithm, Vector Quantization, Simply Genetic Codebook Algorithm

1. INTRODUCTION

The image compression is a very important technology in both communication and computer science. In the past decades, many researches have been increased with multimedia and network in which they have been focused on how to improve the quality of images, to decrease the bit rate and distortion rate. The image compression technology can be divided into two fields. One is loss compression, the other is lossless compression. The loss compression is to reduce the amount of storing materials and get a better compression ratio. For this reason, the restore image is going to produce the distortion after compressing substantially. The lossless compression will not cause distortion problem, but low level compression ratio.

The goal of the image compression is to reduce the amount of the image information data that representing the digital images. The advantage that has been done in this way can reduce the transition time and storage space. Basically, the digital images can be divided into the spatial domain and frequency domain. Before compressing the image data of the frequency domain, we must transform the image data of the spatial domain into the frequency domain, for example, using Discrete Cosine Transform [1] or Hadamard Transform [11].

Vector quantization [10] is one of the most used ways to compress the image data in the spatial or frequency domain. Vector quantization consists of encoding, compressing and decoding. Digitalizing image must take two processes of sampling and scalar quantization. VQ and SQ have some kinds of similarity in the definition. Both of them take one value that represents all possible values in the fetching value blocks, so that the quantization errors can’t be avoided. Comparing to SQ that takes one value representing all possible ones in the fixed or unfixed fetching blocks, VQ takes the results of SQ as a set. It takes one vector as a set of similar vectors that representing all possible vectors in the fetching blocks. The vector is called code vector, and the set of code vectors is called codebook. Hence, in the decoding process of VQ, the level of codebooks will affect the quality of restoring images. In the encoding process of VQ, the codebook generation is the core procedure. Lloyd [15] and Gray [14] bring up a basic method, called LBG algorithm, which can compute the errors between the training sample and the
generated codebook. It will complete the whole training processes of codebooks to get a better one through thousands of computation and comparison. Genetic algorithm is a kind of optimization and widely used in neural networks, fuzzy systems and machine learning, especially suitable for looking for the optimal solution in the vast solving space. That is why we use the advantage of genetic algorithms (GA) to look for a best codebook in accordance with the training sample.

The structure of this paper is as follows: Section two sketch the genetic algorithm and its structure; Section three illustrate and analyze the VQ structure; Section four the research structure and experiment method; Section five the experiment result; Section six the conclusion and research direction in the future.

2. GENETIC ALGORITHMS

In the 19th century, Biologist Darwin put forward the Darwin’s Theory of Evolution, who thought the changing environment will make random mutation to some species, so that they can adapt to the new environment and take the advantage of living competition. During 1950 to 60, the evolutionary concept in the biology has inspired many researchers to use the computer science technology for solving specific problems and the field of the artificial intelligence. Therefore, it evolved the Evolutionary Programming subject, including the Evolution Strategy, Evolutionary Programming, Genetic Programming, and the most representative Genetic Algorithms.

In 1965, while studying the optimization of controllable real number parameters of the hydrodynamics models, Renchenberg put forward an experimental evolution strategy method. After that, he cooperated with Schewefel and solved the numerical problems of non-linear models with probabilities through the evolutionary method that is called Evolution Strategy. In 1966, Fogel took the finite-state machine to represent its behavior in order that the artificial system can adapt itself through evolutionary method and developed a method called Evolutionary Programming.

Between 1960 and 70, Holland and his students devoted the research of evolutionary computation. Initially, they focused on using the artificial system to imitate the mechanism that nature evolves. By 1975, Holland formally proposed the theory structure of Genetic Algorithm [12].

On later stage of 1980, many researchers probed into the genetic algorithms and introduced its applications. For example, Goldberg [9] thought that the genetic algorithm is a search algorithm based on genes and nature selection. He proposed the Roulette Wheel method which uses the concept of the proportions of the rim plate, calculating the proportion levels of the individual fitness to determine the numbers of the reproduction of the offspring in the ethnicity.

The evolutionary computation focused on the Automatic Programming, that is to say which uses the computer program language to solve the problems in early days. Cramer, Fujiki and Dickison applied the genetic algorithms into the automatic programming [17]. In 1992, Koza [13], who made the bit unit of the genetic algorithm spread to the computer program unit, put forward the Genetic Programming which performed the artificial intelligent evolution and problem solving based on the computer programs. Michalewicz [16] thought that we need the basic rules of the traditional GA while problem solving used GA to revise the problem with the proper GA model. For example, binary string encoding, etc. So he submitted a method, using the Data Structure transformation with regard to the problem corresponding to chromosome unit and solving with proper genetic operators, which is called Evolution Programs.

Since the initial stage of 1990, the researchers have studied how to raise the GA efficiency while solving problems. Let's give some examples; Petty [18] proposed that the Parallel Genetic Algorithms (PGA) perform the GA operation with the division of the ethnicity respectively to improve the efficiency of solving. Cantu-Paz [2, 3] and Cantu-Paz, Goldberg [4] studied the degrees of parameters of the parallel genetic algorithms, and they focused on how to design and apply the PGA procedures and parameter setting which can raise the efficiency of solving while solving the single population and multiple population problems.

At present, it has begun to use the GA to solve the problems of the real world in quite a lot fields. For example, Davidor [7] applied the GA on the optimization of the Robot Trajectory. Chen [5] and Croce [6] proposed the application of scheduling.

The goal of GA is to built an artificial genetic system with natural characteristic and makes each species eliminate through competition, that only good adaptable species can survive and propagate to produce the coming generation and leave the species having strong adaptability through the duplication, crossover and mutation repeatedly among a certain living environment. Holland said that the GA is formed by lots of chromosome with each population and uses a method called Natural Selection which lets the old generation chromosomes of population produce the new one with the operators of Crossover, Mutation and Inversion. Mitchell [17] thought the GA including three basic operators that are Selection, Crossover and Mutation, and he proposed a basic step of GA called Generation as follows:
W. H. A. Wang et al.: Codebook Design for Vector Quantization Using Genetic Algorithm

1. Randomly generated \( n \) populations which include \( l \)-bit chromosomes.
2. Calculated the fitness value of each chromosome in the population.
3. Select a pair of parent chromosomes with high fitness value from the current population randomly. The chromosome which owns the higher fitness value could be selected to generate the next generation.
4. With probability \( Pc \) (the crossover rate) decide how many chromosomes of parent population should be crossed over. Cross over the pair at a randomly chosen point (chosen with uniform probability) from two offspring. There are two crossover methods can be used: single-point and multi-point. The numbers of chromosomes being crossed over are \( Pc \times \text{Population size} \).
5. Mutated the two offspring at each locus with probability \( Pm \) (the mutation rate), and place the new-generation chromosomes in the new population. The numbers of chromosomes being mutated are \( Pm \times m \times \text{Population size} \).

3. VECTOR QUANTIZATION

The vector quantization is one of the primary image compression techniques while compressing the image data in the spatial or frequency domain. It is so called the Loss Digital Image Compression in which the distortions have occurred during the restored image procedure and the restored image will differ from the original one after compression procedure. In VQ structure, the process of set of training sample \( \{X_i, i=1,2,\ldots,n\} \) corresponding to the set of codeword \( \{Y_j, j=1,2,\ldots,m\} \) is called VQ encoding, and the set of codeword is called codebook. The goal of VQ encoding is to generate one index of training sample among codebooks which represents the smallest Euclidean distance named between codeword and training vector:

\[
d(X,Y) = \sqrt{\sum_{i,j}(x_i - y_j)^2}
\]  

The average distortion rate of VQ can be computed by the Mean Square Error (MSE):

\[
\text{MSE} = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} d^2(X_i, f(X_i))
\]  

Where, \( N \) denotes the numbers of the training sample set and \( M \) represents the numbers of the codewords set. \( f(X_i) \), a mapping function, maps the training vector \( X \) to the represented codeword of codebook \( Y \). Using mapping function can define the clustering of training vectors and the MSE can find out the nearest or most representative vector of codebook. The basic process flow of VQ is Codebook Generator, Vector Encoder and Vector Decoder as follows:

Step 1: Codebook generating: Depend on the selected way to generate codebooks.
Step 2: Encoding:
1. Cut the original image into regular size image blocks.
2. Calculate and find out the codeword most closing to the image block of codebook. After that, record its index.
3. Output the indexes to a file.
Step 3: Decoding:
1. Make the recorded indices correspond to its codebook, and take out the code words.
2. Restore the codeword to the position of original training sample one by one until the original image shows up. The procedure of encoding and decoding of VQ is as follows:
\[
\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\sum_{i=1}^{W} \sum_{j=1}^{H} (X_{ij} \hat{X}_{ij})^2} \right)
\]  

Where, \(X_{ij}\) and \(\hat{X}_{ij}\) represent the original image pixels and the restored image pixels of coordinate \((i, j)\) respectively. \(W\) and \(H\) represent the width and height of image respectively.

4. RESEARCH FRAMEWORK AND EXPERIMENTAL METHOD

The codebook generation can be regarded as a searching problem, and its goal is to search an optimal solution as the most representative codebook which could correctly be applied in the image compression. We have to meet the following two conditions during the clustering process of VQ:
1. Selected strategies for generating the best codebook. That is the operation algorithm.
2. Standards of the most representative codebook.

Hence, the traditional GA will be used to execute the searching procedure for finding an optimal codebook.

In our research framework, the gray scale image becomes the source of training samples. That is, the image will be cut into the regular size image blocks as training samples. The most representative image block will be selected from the training samples. After classification and selection procedure, the most representative image blocks will be fined out and grouped into a codebook. Therefore, we propose an algorithm which called Simply Genetic Codebook Algorithm (SGCA) to generate the optimal codebook.

SGCA algorithm is as follows:

SGCA structure adopts the real number encoding way and takes chromosome represented codebook as one solution. It also generates an initial population grouped by the training vectors of image blocks. The initial population can be regarded as a feasible solution via the calculated of the evolution of GA. In this paper, we take the PSNR as the fitness value under a setting measurable level and use the selection, crossover and mutation operator to evolve a optimal codebook under a setting logic condition. The experimental steps of SGCA are as follows:

Step1: Pre-handle the original image. Load a gray scale image into a computer memory and cut the image into \(n\) training samples. \(\{T_i, i = 1, 2, 3, \ldots, n\}\)

Step2: Encode procedure. The real numbers are adapted to the codebook generation of chromosomes, in which each chromosome is organized by \(m\) genes.

Step3: Define the PSNR as the fitness value.

Step4: Initial population generated. The initial populations of \(r\) chromosomes are generated from \(n\) training samples with a random selection method. \(\{S_j, j = 1, 2, \ldots, r\}\)

Step5: Evaluation. Select the best chromosome \(S_{best}\), we use PSNR as fitness value and evaluate each chromosome of generation.

Step6: Selection method. Retain the best fitting chromosome of one generation and select the fixed numbers of parent chromosomes \(S_p\) by using the Roulette Wheel method [16].

Step7: Cross over method. Use two-point crossover method in the step and set crossover rate with 100% to generate the new generation. After the procedure of fitness function evaluation, the best fitting children chromosomes can be retained.

Step8: Mutation. Mutate all of children chromosomes \(P \times m \times \text{Population - size}\) times except the best fitting one.

Step9: Terminated condition. Terminate the whole algorithm, if the PSNR is greater than 30 or continual generations of 20 have the same fitness value.

We explain our research method as below. First, pre-handle the original image simply, that is, we cut the original image into \(k \times k\) regular size image block called training vector as same dimension as the original image. Take one 256×256 gray scale image as an example. We cut the image into 4×4 size image...
block and can get 4096 image blocks.

To continue, encoded procedure is executed. In
gray scale image, each pixel ranges 0 to 255. To
search for the most representative codebook in
the space of training vector set grouped by image blocks,
this paper adopts real number encoding, that is, takes
the real numbers of pixels to encode. The length of
the chromosome is the quantity of divided image
blocks N. Each gene of chromosome corresponds
to one image block and be represented by “bit” data
structure. The gene of code word in the codebook is
selected from image blocks directly. For example,
one 256×256 size of image can be cut into 4096 of
4×4 image blocks (called gene). Since we set the
codebook size to 256, the chromosome includes 256
genes of 16 bits length and the size of chromosome is
4096 bits.

The goal of fitness function in this paper is to
find out the most representative m image blocks
forming into the codebook from n training vectors
set. Hence, we have to compute the similarity
\( E[a(X,Y)] \) between training vectors set \( X \)
and codewords set \( Y \) as equation 1 shown. The shortest
length represents the highest similarity, and we can
group the training vector \( X \) and codeword \( Y \) into
one cluster. Computed the distance of training vector
\( X_i \) and codeword \( Y_j \) one by one, we can classify
each training vector into a specify cluster, that is,
used the complete matching method to find out the
nearest codeword \( \hat{X}_i, n = 1, 2, \ldots, n \). To acquire the
highest similarity between the original image blocks
and codebook, PSNR evaluation method is adopted in
this paper. The objective function \( F(\hat{X}) \) is defined
as bellows:

\[
F(\hat{X}) = 10 \sum_{j=1}^{n} \sum_{i=1}^{m} \log_{10} \left( \frac{255^2}{\text{MSE}_X} \right)
\]  

(4)

Where, \( n \) and \( m \) denote the numbers of horizontal and
vertical training samples of an image.

After the encoding procedure, \( \hat{X} \) denote
chromosome \( S_i \), the bigger is the objective value
of \( \hat{X} \), the higher is the fitness value of
chromosome \( S_i \). Therefore, we use PSNR as the
fitness function directly. After defining the
chromosome encoding and fitness function, each
chromosome can be seen as a feasible solution and
evolved by GA to find the highest fitness
chromosome, which is the most representative
codebook.

SGCA uses the operators of selection,
crossover and mutation as bellows:
1. Initial population generated: In the initialization, \( S \)
denotes the population size, and \( M \) denotes the
codebook size, and \( N \) denotes the size of numbers
of training vector. We randomly generate \( S \) chromosomes grouping into an initial population,
in which each chromosome is an individual by \( M \)
genomes and represents a codebook with
\( C_M^n \) combination.
2. Selection: Used the Breeder Selection method
compute the fitness value for each chromosome in
one generation. Keep the highest chromosome \( S_{\text{best}} \),
and use the Roulette Wheel method which selects
fixed \( S_b \) numbers of parent chromosomes to
generate the children.
3. Duplication: The parent chromosomes of each
generation population are duplicated and stored in
the crossover pool directly.
4. Crossover: Used two-point method and set the
crossover rate with 100%. First, we run
omously generate \( N - 1 \) crossover points distributed and randomly generate
of crossover. \( N \) genes uniformly generate \( N \)-1
crossover pool directly.

- \( S_i \) is the parent chromosome
- \( S_j \) is the parent chromosome
- \( S_i' \) and \( S_j' \) after
crossover

5. Mutation: The default value of mutation rate is
0.01. As for the children generation after crossover,
compute the fitness of each chromosome in one
generation. Retain the highest one, but mutate the
others. To complete the mutation process, the
process is randomly select one chromosome and
one gene of it; arbitrarily choose one image block of
the original image to replace the gene.

SGCA algorithm shows as Figure 6. In order to
meet traditional GA logically, we set some conditions
as follows:
1. Each chromosome combines \( M \) image blocks
which are randomly selected from N image blocks in the initial population. To avoid the duplication problem of genes in one chromosome, we set there are not same genes in a chromosome.

2. Since the crossover rate is 100%, we randomly select two parent chromosomes to crossover in the pool. To avoid the duplication selection problem of chromosomes, we use Without Replacement method to complete the crossover procedure.

3. To make chromosomes change obviously in each generation after genetic algorithm, we retain the highest fitness chromosome, but mutate the others in each generation. Besides, we take the training sample as the substitution of mutation $MRV = \{S_i, S_j, \ldots, S_n\}$ and group them into three clusters 

$$MRV_1 \subseteq MRV_2 \subseteq MRV_3.$$ 

Where, $MRV_i$ represents the nearest average mutation substitution, and $MRV_i = MRV$. Hence, if generating the same PSNR of 10 continuous generations, replaces the mutation range by $MRV_i$. And so forth, make sure the reachability of GA.

5. EXPERIMENTAL RESULTS

The experiment platform of this paper is Pentium IV 2.0 GHz processor, 256 MB RAM, Windows 2000 Professional operating system and MATLAB 6.0. We use two 256×256 gray scale images of Lena and Barbara to complete the experiment as Figure 7 and 8 shows.

In the image parameter setting, the size of training sample is 4×4, and the codebook size is 256. In genetic algorithm, the initial population size is 100, and the numbers of parent population are 10 in the pool. The crossover rate is 100% and the mutation rate is 0.01. Therefore, each generation has to be mutated 51 times (0.01×256×20 ÷ 51). We stop the searching function when meets the termination condition in which the fitness of PSNR reaches 30 or the same fitness value of the best chromosome during continuous 20 generations. At this time, the best PSNR is the optimal solution of this experiment.

The main purpose of this experiment is to apply GA on the design of codebooks in vector quantization, and finds out the best codebook through SGCA structure, in which the running time and efficiency are not in consideration. The comparison of SGCA, LBG, DCT+LBG and HT+LBG [19] is as table 1 shows. In the experiment results, SGCA can generate higher PSNR value than LBG, in which SGCA provide a GA-base algorithm to search an optimal codebook which acquire better restoring quality.

6. CONCLUSION AND FUTURE PROJECT

In this variety information era, how to effectively compress mass information is becoming a major issue. Nowadays, Image compression technology becomes more and more important and prevalent. In this paper, we apply genetic algorithms based on the natural evolution rules to search the best codebook in vector quantization. After evaluating the image restoring quality, SGCA gets a better result.
than LBG. To speed up evolution process, the crossover rates are tuned first and the chromosome encoding method is modified, in which to get higher evolution efficiency even more imbed stochastic model to SGCA in the future.

REFERENCES


ABOUT THE AUTHORS

Wei-Hua Andrew Wang received his M.S. and Ph.D. in Industrial and Systems Engineering from Ohio State University, and B.S in Industrial Engineering from Tunghai University in Taiwan. He was on the faculty of the Information Management and Technology Department of National Taiwan University of Science & Technology and has been on the faculty of industrial engineering at Tunghai University since 1992. He was also a Visiting Associate Professor at Ohio State University in 1998-1999. Dr. Wang’s expertise is in the areas of knowledge management, business process analysis, and applied artificial intelligence in production system planning. He has extensive industry and government consulting experience and has been a Principle Investigator of more than 20 research projects funded by the National Science Council of Taiwan, major steel and machine tools manufacturers, aerospace firms, and hospitals in Taiwan. Most recently, he has successfully founded an interdisciplinary industry research center (Center for Collaborative Business) at Tunghai University with multiple-year funding from Taiwan’s Ministry of Education.

Chun-Liang Tung received the MS degree in Computer Information Science from Knowledge System Institute in 1997. Mr. Tung currently is a lecturer in the Department of Information Management at the National Chin-Yi Institute of Technology, Taiwan. He is a Ph.D. graduate student of Industrial Engineering and Enterprise Information at Tunghai University. His research interests include image compression, information security, artificial intelligence, and electronic commerce.

Shih-Chun Yang received the BS degree in Industrial Engineering from Tunghai University in 1999. Mr. Yang currently is a postgraduate student of Industrial Engineering and Enterprise Information at Tunghai University. His interests include healthcare quality, hospital management, and System Engineering.

(Received July 2004, revised September 2004, accepted November 2004)