A kernel vector quantization codebook designing for image compression based on simulated annealing into genetic algorithm

Mengling Zhao [1,2*], Hongwei Liu [1]

[1]*School of Mathematics and Statistics, Xidian University, 710071 ,Xi'an,China*
[2]*School of Science, Xi'an University of Science and Technology, 710000 ,Xi'an,China*

Received 1 December 2014, www.cmnt.lv

Abstract

To solve premature phenomenon and falling into local optimum of genetic algorithm, the simulated annealing algorithm is introduced to the genetic algorithm and a simulated annealing is presented based on genetic clustering algorithm, a new effective SA, crossover operator and mutation operator proposed for fitting the partition-based chromosome coding. In addition, the Euclidean distance is replaced by the kernel function distance to improve the performance of the proposed algorithm further. We also applied the proposed algorithm to image compression. Experimental results indicate its superiority in terms of peak signal to noise ratio (PSNR).

Keywords: Mercer Kernel; crossover operator; mutation operator; Vector Quantization

1 Introduction

The main purpose of Image Compression is as much as possible to reduce the amount of data image compression. Vector quantization (VQ) is one of the most important technologies for image compression [1-3], and VQ has been widely used in different applications such as image compression [4-6], pattern recognition [7-9], speech coding [10, 11], etc. In general, VQ utilizes the same codebook at the sending terminal and the receiving terminal respectively to encode and decode the signal. In the encoding phase, VQ finds the nearest codeword for each input vector according to a certain distortion criterion, and transmits the corresponding indexes to the decoder. Thus, in the decoding phase, merely a simple table-look-up operation is required.

A vector quantizer Q maps the K-dimensional space $\Re^K$ into a finite subset of $\Re^K$ and statistically encodes data vectors in order to quantize and compress the data. Based on the centroid condition and the nearest neighbor condition, Linde et al. [12] proposed the popular Linde-Buzo-Gray (LBG) algorithm. However, LBG algorithm strongly depends on the choice of the initial codebook and cannot gain the best solution. Some improved versions of LBG have been proposed, such as the Enhanced LBG (ELBG) [13], Adaptive Incremental LBG (AILBG) [14], Evolutionary-based Methods [15, 16], Mercer kernel methods [17], etc.

In recent years, various bio-inspired optimization techniques have been widely used in varieties of optimization tasks. Genetic Algorithms (GA), first pioneered by John Holland [18] in the 1970s, is an efficient and global optimum search method based on the ideas of nature selection and genetics. Its main idea is to maintain a set of solutions (population), which is regenerated iteratively using genetic operations (crossover and mutations) and selection [19]. GA has been proven to be powerful methods and widely used in many fields.

Simulated Annealing (SA) is a global optimum search method based on the idea of physical annealing and has higher convergence speed than GA. However, the initial temperature and the decreasing speed of the annealing temperature have a great effect on the performance of SA algorithm. If the initial temperature is high enough and the annealing temperature decreases slowly enough, the obtained solution approaches the global solution with probability one but the convergence speed is very low. On the other hand, SA algorithm is easy to fall into local optimum if the annealing temperature decreases quickly [20].

The remainder of this paper is organized as follows. Section 2 describes some related background such as the concept of vector quantization, the LBG algorithm and the Mercer kernel. Section 3 presents the proposed algorithm in detail. Section 4 shows the experimental results. Finally, a conclusion is given in Section 5.

2 Background

In this section, the concept of vector quantization, the LBG algorithm and Mercer kernel are introduced simply.

* Corresponding author's E-mail: zhaomengling@126.com

2.1 VECTOR QUANTIZATION

A vector quantizer Q maps vector set $\mathrm{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N\},\boldsymbol{x}_i\in\Re^K$ into a finite set of representative vectors $V=\{\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_M\}$ and divides the vector space into M disjoint regions $C_1,C_2,\ldots,C_M$, and

$$C_1\cup C_2\cup\ldots\cup C_M=\Re^K \text{ and } C_i\bigcap C_j=\Phi,i\neq j \quad (1)$$

Where,

$$C_i=\{\boldsymbol{x}\in\Re^K:Q(\boldsymbol{x})=\boldsymbol{v}_i\},i=1,2,\ldots,M \quad (2)$$

The representative vector set V is referred as codebook and the representative vector $\boldsymbol{v}_i$ is called codeword. The goal of designing optimal vector quantizer is to seek the optimal codebook to minimize the distortion of vector quantizer.

If the squared error $d(\boldsymbol{x},Q(\boldsymbol{x}))=\|\boldsymbol{x}-Q(\boldsymbol{x})\|^2$ is adopted as the distortion metric, the distortion function of vector quantizer can be expressed as:

$$\mathrm{D}(X,V)=\sum_{i=1}^{M}\sum_{k\in C_i}\|\boldsymbol{x}_k-\boldsymbol{v}_i\|^2 \quad (3)$$

VQ can be regarded as a clustering problem, in which an assignment of vectors to clusters is desired such that distortion function of the vectors to their cluster centroid is minimal.

2.2 LBG ALGORITHM

In general, the LBG algorithm iteratively applies the nearest neighbor condition and the centroid condition to produce the optimal codebook. The LBG algorithm consists of the following steps:

Step1: Initialization. Given an initial codebook $V^{(0)}=\{\boldsymbol{v}_1^{(0)},\boldsymbol{v}_2^{(0)},\ldots,\boldsymbol{v}_M^{(0)}\}$; set $t=0$, $\mathrm{D}^{(-1)}=+\infty$ and $\varepsilon\geq 0$;

Step2: Partition. Given codebook $V^{(t)}=\{\boldsymbol{v}_1^{(t)},\boldsymbol{v}_2^{(t)},\ldots,\boldsymbol{v}_M^{(t)}\}$, assign each training vector to the corresponding region (Voronoi cell) according to the nearest neighbor condition, that is,

$$C_i=\{\boldsymbol{x}_k\in\Re^K:\|\boldsymbol{x}_k-\boldsymbol{v}_i\|^2\leq\|\boldsymbol{x}_k-\boldsymbol{v}_j\|^2\ \forall j\neq i\},i=1,2,\ldots,M \quad (4)$$

and compute the distortion using Eq. (3), that is

$$D^{(t)}=\sum_{i=1}^{M}\sum_{k\in C_i^{(t)}}\left\|\boldsymbol{x}_k-\boldsymbol{v}_i^{(t)}\right\|^2$$

Step3: Terminal condition check. If $\left(\mathrm{D}^{(t-1)}-D^{(t)}\right)/D^{(t-1)}\leq\varepsilon$, then stop and output $V^{(t)}$ which represents the final codebook; else, continue;

Step4: Codebook update. Calculate the new codeword according to the centroid condition, that is,

$$\boldsymbol{v}_i^{(t)}=\frac{1}{\left|C_i^{(t)}\right|}\sum_{x_j\in C_i^{(t)}}\boldsymbol{x}_j,i=1,2,\ldots M \quad (5)$$

Step5: Set $t=t+1$ and go to Step2.

2.3 MERCER KERNEL

Given data set $\mathrm{X}=\{x_1,x_2,\ldots,x_N\},x_i\in\Re^K$ and a nonlinear mapping function $\Phi(\bullet)$ that maps x_i from the input data space $\Re^K$ to a new feature space F with higher or even infinite dimensions, the dot product of $\Phi(x_i)$ and $\Phi(x_j)$ in feature space F can be implicitly computed by using the Mercer kernel function $K(x_i,x_j)$ [17] defined in the input space, i.e. $K(x_i,x_j)=\langle\Phi(x_i),\Phi(x_j)\rangle$ without needing the knowledge about the specific form of the nonlinear mapping $\Phi(\bullet)$. Therefore, any computations in the feature space F can be efficiently converted into operations in the data space through the kernel function.

In this paper, the Gaussian kernel function is introduced and it is given as follows:

$$K(\boldsymbol{x},\boldsymbol{y})=\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{y})^2}{\sigma^2}\right) \quad (6)$$

According to the Gaussian kernel function $K(\boldsymbol{x},\boldsymbol{y})$ and the Euclidean distance $d(\boldsymbol{x},\boldsymbol{y})$, we observe the connection between them as follows: $K(\boldsymbol{x},\boldsymbol{y})$ is inversely proportional to $d(\boldsymbol{x},\boldsymbol{y})$. For example, if the value of $d(\boldsymbol{x},\boldsymbol{y})$ is the lower bound 0, $K(\boldsymbol{x},\boldsymbol{y})$ reaches the upper bound 1; and on the contrary, if $d(\boldsymbol{x},\boldsymbol{y})$ approaches infinity, $K(\boldsymbol{x},\boldsymbol{y})$ is closed to the lower bound 0. It is the reason why we use the Gaussian kernel function rather than other kernel functions. Further, many image data are subjected or approximately subjected to Gaussian distribution in real world.

So, the Euclidean distance in input space can be replaced by the Gaussian kernel function:

$$d(\boldsymbol{x}_k,\boldsymbol{v}_i)=2\cdot(1-K(\boldsymbol{x}_k,\boldsymbol{v}_i))=2\cdot\left(1-\exp\left(-\frac{(\boldsymbol{x}_k-\boldsymbol{v}_i)^2}{\sigma^2}\right)\right) \quad (7)$$

When $\boldsymbol{x}_k$ is closer to $\boldsymbol{v}_i$, $d(\boldsymbol{x}_k,\boldsymbol{v}_i)$ becomes smaller until it reaches zero. In reverse, when $\boldsymbol{x}_k$ is far away from $\boldsymbol{v}_i$, $d(\boldsymbol{x}_k,\boldsymbol{v}_i)$ will get larger until it reaches 2 (its maximum value). In Eq. (7), the function of constant 2 aims to extend the range of $d(\boldsymbol{x}_k,\boldsymbol{v}_i)$.

The total distortion of vector quantizer in space can be expressed in Eq. (8):

$$D=\sum_{i=1}^{M}\sum_{k\in C_i}2\cdot(1-K(\boldsymbol{x}_k,\boldsymbol{v}_i)) \quad (8)$$

3 Kernel Vector Quantization Based on Simulated Annealing into Genetic

In this section, kernel vector quantization based on simulated annealing into genetic (KVQSAG) algorithm is presented in detail. Vector quantization based on simulated annealing into genetic (VQSAG)algorithm mainly consists of the operators of GA and the simulated annealing method, and the following will introduce each operation employed in the VQSAG.

3.1 PARTITION-BASED CODE DESIGNING

In the partition-based code scheme, the genes in each individual are the labels of training vector and the labels represent the clusters which the training vector belong to. The *k-th* gene in individual represents the label of the *k-th* training vector. A group of labels consist of an individual. Suppose that the number of the training vector is *N*, then the length of individual is also *N*. Therefore, each individual represents one division of the training vector, and the goal of the operators in GSAVQ algorithm and GSAKVQ algorithm is to seek the optimal division of the training vector to minimize the total distortion of vector quantizer.

The individual A_c can be expressed as follows:

$$A_c = \{a_{c1}, a_{c2}, ..., a_{ck}, ..., a_{cN}\} \tag{9}$$

Where, $a_{ck} \in \{1, 2, ..., M\}$ is the *k-th* gene in individual A_c and it represents the label of the *k-th* training vector. *M* is the size of codebook or the number of labels.

The initial population with P individuals can be stated as follows:

$$A = \{A_1, A_2, ..., A_c, ..., A_P\} \tag{10}$$

3.2 FITNESS FUNCTION

In this paper, the fitness of individuals in the feature space is proposed on the basis of Ref. [21].

The distortion function of vector quantizer in the kernel feature space is shown in Eq. (8), and we can rewrite it as follows:

$$D = \sum_{i=1}^{M} \sum_{k \in C_i} 2 \cdot (1 - K(\boldsymbol{x}_k, \boldsymbol{v}_i)) = \sum_{i=1}^{M} d_i \tag{11}$$

Where, d_i is the local distortion of vector quantizer:

$$\begin{aligned} d_i &= \sum_{k \in C_i} 2 \cdot (1 - K(\boldsymbol{x}_k, \boldsymbol{v}_i)) = 2 \cdot |C_i| - 2 \cdot \sum_{k \in C_i} K(\boldsymbol{x}_k, \boldsymbol{v}_i) \\ &= 2 \cdot |C_i| - 2 \cdot \sum_{k \in C_i} K\left(\boldsymbol{x}_k, \frac{\sum_{m \in C_i} \boldsymbol{x}_m}{|C_i|} \right) = 2 \cdot |C_i| - \frac{2}{|C_i|} \cdot \sum_{k \in C_i} \sum_{m \in C_i} K(\boldsymbol{x}_k, \boldsymbol{x}_m) \end{aligned} \tag{12}$$

Set $s_i = \frac{1}{|C_i|} \sum_{k \in C_i} \sum_{m \in C_i} K(\boldsymbol{x}_k, \boldsymbol{x}_m)$, then the local distortion can be written as follows:

$$d_i = 2 \cdot |C_i| - 2 \cdot s_i \tag{13}$$

So, the total distortion of vector quantizer is given as follows:

$$D = \sum_{i=1}^{M} (2 \cdot |C_i| - 2 \cdot s_i) = 2N - 2 \cdot \sum_{i=1}^{M} s_i \tag{14}$$

From Eq. (14), it is clear that minimizing the total distortion of vector quantizer is equivalent to maximizing the second term of Eq. (14). So, the fitness function of individuals is defined as follows:

$$fitness = 2 \cdot \sum_{i=1}^{M} s_i \tag{15}$$

3.3 SIMULATED ANNEALING METHOD

By performing generic simulated annealing on one individual, a new individual is generated and join into evolution with certain probability into the next generation. The method does not take into account the effect of each gene on the distortion of vector quantizer. In the proposed KVQSAG, an effective simulated annealing method is proposed based on an analysis of effect of each gene on the distortion.

When the values of genes in an individual are changed, the total distortion of vector quantizer can be given as follows:

$$\hat{D} = 2 \cdot \left(N - \sum_{i=1}^{M} \hat{s}_i \right) \tag{16}$$

According to $s_i = \frac{1}{|C_i|} \cdot \sum_{k \in C_i} \sum_{m \in C_i} K(\boldsymbol{x}_k, \boldsymbol{x}_m)$, the distortion variation will be:

$$\hat{D} - D = -2 \cdot \left(\sum_{i=1}^{M} \hat{s}_i - \sum_{i=1}^{M} s_i \right) \tag{17}$$

According to

$$\begin{aligned} \hat{D} - D &= -2 \cdot \left(\sum_{i=1}^{M} \hat{s}_i - \sum_{i=1}^{M} s_i \right) \\ &= -2 \cdot \left(\left(\hat{s}_{a_k} + \hat{s}_j \right) - \left(s_{a_k} + s_j \right) \right) \end{aligned} \tag{18}$$

Where

$$\hat{s}_{a_k} = \frac{1}{|\hat{C}_{a_k}|} \cdot \sum_{k \in \hat{C}_{a_k}} \sum_{m \in \hat{C}_{a_k}} K(\boldsymbol{x}_k, \boldsymbol{x}_m) = \frac{s_{a_k} \cdot |C_{a_k}| - 2 \cdot \sum_{m \in C_{a_k}} K(\boldsymbol{x}_k, \boldsymbol{x}_m)}{|C_{a_k}| - 1},$$

$$\hat{s}_j = \frac{1}{|\hat{C}_j|} \cdot \sum_{k \in \hat{C}_j} \sum_{m \in \hat{C}_j} K(\boldsymbol{x}_k, \boldsymbol{x}_m) = \frac{s_j \cdot |C_j| + 2 \cdot \sum_{m \in C_j} K(\boldsymbol{x}_k, \boldsymbol{x}_m) + 2}{|C_j| + 1}$$

Therefore, the *k-th* gene whose value is changed from a_k to *j*, the distortion variation will be:

$$\begin{aligned} \Delta_j^k &= \hat{D} - D \\ &= -2 \cdot \left(\left(\frac{s_{a_k} \cdot |C_{a_k}| - 2 \cdot \sum_{m \in C_{a_k}} K(\boldsymbol{x}_k, \boldsymbol{x}_m)}{|C_{a_k}| - 1} + \frac{s_j \cdot |C_j| + 2 \cdot \sum_{m \in C_j} K(\boldsymbol{x}_k, \boldsymbol{x}_m) + 2}{|C_j| + 1} \right) - \left(s_{a_k} + s_j \right) \right) \end{aligned} \tag{19}$$

3.4 CROSSOVER OPERATION

The genes in an individual represent the labels of the training vector. a_{ik} and a_{jk} are the *k-th* gene in the *i-th* individual and in the *j-th* individual respectively, and they represent the labels of the *k-th* training vector. Generally,

a_{ik} is not equal to a_{jk}, however, a_{ik} and a_{jk} represent the same cluster in essence. In other words, different individuals use different labels to represent the homogenous group of the training vector. Therefore, if the single point-crossover or multiple-point crossover operator is adopted, the excellent pattern in individual may be destroyed and no good results can be obtained. So, based on the special partition-based code scheme used in KVQSAG, a novel special and effective crossover operation is designed and it is introduced in the following.

Suppose individual A_i and individual A_j are chosen. The crossover operation is done as follows: (a) for individual A_i, randomly select one label *rand* and the genes whose values are equal to *rand* are recorded as collection C^i_{rand}; (b) for individual A_j, find out the distribution $cnt^k_{C^i_{rand}}$ that represents the number of genes whose values are equal to *k* in the corresponding position of C^i_{rand}; (c) set $u = \max_{k=1,2,..,M}\left\{cnt^k_{C^i_{rand}}\right\}$ and record the genes in individual A_j whose values are equal to *u* as collection C^j_u; (d) set the values of genes in individual A_i in the corresponding position of C^j_u to *rand* and set the values of genes in individual A_j in the corresponding of C^i_{rand} to *u*.

3.5 MUTATION OPERATOR

The mutation probability is set as P_m. Once an individual is chosen, we compute the local distortions $d_i, i = 1,2,...,M$, and select one of the local distortions d_i by using the roulette strategy according to the local distortions. Then determine the gene *r* in collection C_i which influences the distortion d_i most and change the value of the gene *r* to a randomly chosen label $j, i \neq j$. How to determine the gene *r* is explained as follows.

The local distortion with the Gaussian kernel function is as follows:

$$d_i = |C_i| - \frac{1}{|C_i|} \cdot \sum_{j\in C_i}\sum_{m\in C_i} K\left(\boldsymbol{x}_j, \boldsymbol{x}_m\right) \quad (20)$$

If one of the genes *k* in collection C_i is changed, the local distortion d_i can be expressed:

$$d_{i/k} = \left(|C_i|-1\right) - \frac{1}{|C_i|-1} \cdot \sum_{j\in C_i, j\neq k}\ \sum_{m\in C_i, m\neq k} K\left(\boldsymbol{x}_j, \boldsymbol{x}_m\right) \quad (21)$$

The gene $r = \min_{k\in C_i}\left\{d_{i/k}\right\}$ is the gene that influences the distortion d_i most. From Eq. (21), it is clear that minimizing $d_{i/k}$ is equivalent maximizing the second term of Eq. (21), which is equivalent to minimizing $\sum_{j\in C_i} K\left(\boldsymbol{x}_j, \boldsymbol{x}_k\right)$. So, the gene $r = \min_{k\in C_i}\left\{\sum_{j\in C_i} K\left(\boldsymbol{x}_j, \boldsymbol{x}_k\right)\right\}$ is the gene that influences the distortion d_i most.

3.6 THE FRAMEWORK OF KVQSAG

Based on the introduction of each operation in the previous sections, the framework of KVQSAG is described as follows:

Step1: Initialize the population and the annealing temperature and set $i=0$;

Step2: Compute the fitness of each individual;

Step3: If the stopping criterion is satisfied, stop and decode the best individual as the optimal codebook; otherwise, continue;

Step4: Simulated annealing operation;

Step5: Perform crossover operation, mutation operation and selection operation in turn;

Step6: $i=i+1$; return Step2.

The stop criterion is defined as follows:

$$\left|fitness(best(t)) - fitness(best(t-1))\right| < \varepsilon \quad (22)$$

Here, $fitness(best(t))$ and $fitness(best(t-1))$ are the fitness values of the best individual in *t-th* and t-1-*th* iteration respectively.

4 Application in Image Compression

In this paper, an adaptive method is used to determine the initial temperature for different training vector set.

The adaptive method for initial temperature is given as follows: for the initial population, the average of $\Delta_{i,j} > 0$ is computed, which is used set as the initial temperature.

Other parameters in the proposed algorithm are set as follows: the population has 20 individuals, the mutation probability $P_m = 0.1$, the crossover probability $P_c = 0.7$ and the decreasing speed of the annealing temperature is set to 0.8.

In this experiments, we have taken 20 independent runs (with different seeds of the random number generator) of each algorithm.

All the algorithms discussed here have been developed in a Matlab7.0.1 platform on a Pentium-IV 2.33-GHz PC, with a 2-GB main memory in Windows Server 2003 environment.

4.1 APPLICATION IN IMAGE COMPRESSION

In this section, we validate the effectiveness of proposed approach in image compression. Firstly, the performance of the proposed algorithm is compared with that of LBG [12], Improved Particle Swarm Optimization cluster method (IPSO) [22] and the steady-state memetic algorithm [23] (MA) in terms of the peak signal to noise ratio (PSNR), the running time and the visual effect of recovered images.

Each test problem runs 10 times independently and an average result of the peak signal to noise ratio (PSNR) is recorded. For a $m \times m$ grayscale image, the PSNR is defined as follows:

$$PSNR = 10 \times \lg \left(\frac{255^2}{\sum_{i=1}^{m}\sum_{j=1}^{m}(x_{ij} - \hat{x}_{ij})^2 / m} \right) \qquad (22)$$

where x_{ij} is the original pixel value and $\hat{x}_{ij}$ is compressed pixel value.

Grayscale images "Lena", is used as the training images (as shown in Figure 1) and "boat", "airplane", "einstein" are used as the test images. The size of each image is 512×512.

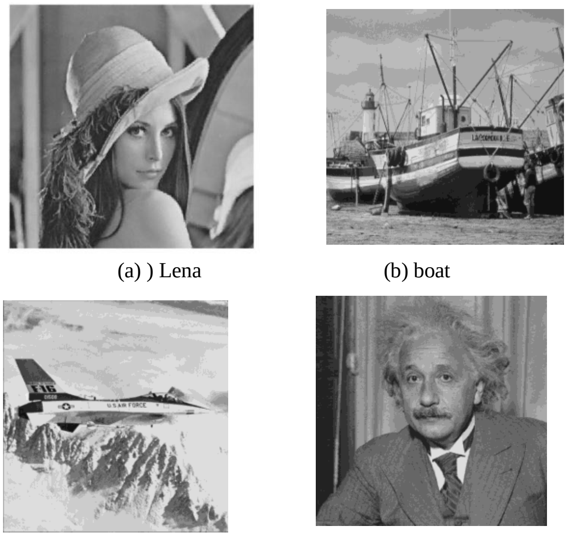
(a)) Lena (b) boat (c) airplane (d) Einstein

FIGURE 1 The training images

Table 1 presents the mean PSNR and table 2 shows the mean running time of four algorithms in the 10 independent runs. The bold in table 1 is the optimal value of the four algorithms.

From Table 1, it is can be seen that the proposed KVQSAG algorithm has a better performance than others in term of PSNR whether in the training image or out of the training image.

TABLE 1 PSNR (dB) comparison of four algorithms with the different codebook sizes

Training Images	Algorithms	Codebook size			
		64	128	256	512
lena	LBG	29.1225	30.2706	31.0375	31.9970
	IPSO	29.3155	30.2771	31.2545	32.1210
	MA	28.9800	30.0628	31.0048	32.1364
	KVQSAG	**29.6624**	**30.7130**	**31.6952**	**32.9009**
boat	LBG	26.3006	27.2585	27.8510	28.6008
	IPSO	26.4740	**27.4326**	27.9996	28.6148
	MA	**26.4920**	27.3817	27.8903	28.6647
	KVQSAG	26.4885	27.3465	**28.1263**	**28.7893**
airplane	LBG	26.6706	27.4460	27.8753	28.5908
	IPSO	26.5978	27.5698	28.0306	28.5094
	MA	26.6249	27.2531	27.8840	28.6896
	KVQSAG	**26.7897**	**27.6241**	**28.2529**	**28.8415**
einstein	LBG	30.8522	31.8297	32.6087	33.1932
	IPSO	31.1296	31.8260	32.6032	33.2318
	MA	30.7521	31.7100	32.5510	33.2385
	KVQSAG	**31.1669**	**31.9828**	**32.7100**	**33.3542**

From Table 2, it is clear that KVQSAG has the largest running time among the algorithms. However, the running time of the KVQSAG algorithm is not changed with the size of codebook. If the data set is small and the size of codebook is large, KVQSAG algorithm can be selected suitably.

TABLE 2 Running time (s) comparison of four algorithms with different codebook sizes

Algorithms	Codebook size			
	64	128	256	512
LBG	9	16	26	52
IPSO	304	621	1285	2725
MA	79	138	278	568
KVQSAG	5139	5065	5220	5116

5 Conclusion

A kernel vector quantization based on simulated annealing **i**nto genetic algorithm is proposed in the paper and it is applied to image compression. In the proposed algorithm, first, a novel simulated annealing method based on an adaptive method for determining the initial temperature is introduced into the genetic algorithm to guide the evolution process so as to make the algorithm escape from the local optimal. Moreover, a novel crossover is specially designed for partition-based coding method. The experimental results show the effectiveness of the proposed method.

Acknowledgement

This work is supported partly by the National Natural Science Foundation of China under an No. 61072144 and No. 61179040. This work is supported partly by the Tianyuan Special Funds of the National Natural Science Foundation of China (Grant No. 11426171).

References

[1] Yeou-Jiunn Chen, Shih-Chung Chen and Jiunn-Liang Wu 2014A Hybrid Vector Quantization Combining a Tree Structure and a Voronoi Diagram, *Mathematical Problems in Engineering*, **2014** , Article ID 758734, 9 pages.

[2] J. Cui, Z. An, Y. Guo, and S. Zhou 2010 Efficient nearest neighbor query based on extended B+-tree in high-dimensional space, *Pattern Recognition Letters* **31**(12), 1740–1748.

[3] S. R. Tsui, C. T. Huang, and W. J. Wang 2013 A new adaptive steganographic method based on gradient adjacent prediction and side-match vector quantization, *Journal of Information Hiding and Multimedia Signal Processing* **4**(4), 215–224.

[4] C. C. Hana, Y.N. Chenb, C.C. Loc, C.T. Wangd 2007 A novel approach for vector quantization using a neural network, mean shift, and principal component analysis-based seed re-initialization. *Signal Processing* **87** (5), 799-810.

[5] R.C. Liu, L.C. Jiao, W. Zhang, J.J. Ma 2010 Clonal selection algorithm for image compression. *In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation, CEC2010, Barcelona, Spain* 2010, 1457-1465.

[6] S. Sathappan 2013 A Vector Quantization Technique for Image Compression using Modified Fuzzy Possibility C-Means with Weighted Mahalanobis Distance. *International Journal of Innovative Research in Computer and Communication Engineering* **1**(1), 1-20.

[7] T. Kohonen 1991 Learning vector quantization for pattern recognition. Report TKK-F-A601, *Helsinki University of Technology*.

[8] E. Lughofer 2008 Extensions of vector quantization for incremental clustering. *Pattern Recognition* **41**(3), 995-1011.

[9] S.X. Chen, F.W. Li 2011 Fast encoding method for vector quantization of images using subvector characteristics and Hadamard transform, *IET Image Processing*, 18–24

[10] K. Krishnaa, V.L.N. Murtyb, K.R. Ramakrishnan 2001 Vector quantization of excitation gains in speech coding. *Signal Processing* **81**(1), 203-209.

[11] S. Chatterjee, T.V. Sreenivas 2008 Optimum switched split vector quantization of LSF parameters. *Signal Processing* **88**(6), 1528-1538.

[12] Y. Linde, A. Buzo, R.M. Gray 1980 An algorithm for vector quantizer design *IEEE transactions on communications* **28**, 84-94.

[13] G. Patané, M. Russo 2001 The enhanced LBG algorithm. *Neural Networks* **14** (9),1219-1237.

[14] F. Shen, O. Hasegawa 2006 An adaptive incremental LBG for vector quantization. *Neural Networks* **19** (5), 694-704.

[15] S.M. Pan, K.S. Chenga 2007 An evolution-based tabu search approach to codebook design. *Pattern Recognition* **40**, 476-491.

[16] C. Azevedo, E. Bispo, F. Tiago, M. Madeiro, M.S. Alencar 2008 An evolutionary approach for vector quantization codebook optimization. *Advances in Neural Networks* **52**(63), 452-461.

[17] D. Bacciu, A. Starita 2009 Expansive competitive learning for kernel vector quantization. *Pattern Recognition Letters* 30, 641-651.

[18] J.H. Holland 1975 Adaptation in natural and artificial systems. *The University of Michigan Press, Ann Arbor, MI.*

[19] P. Fränti 2000 Genetic algorithm with deterministic crossover for vector quantization. *Pattern Recognition Letters* **21**(1), 61-68.

[20] H.C. Huang, J.S. Pan, Z.M. Luc, S.H. Sunc, H.M. Hanga 2001 Vector quantization based on genetic simulated annealing. *Signal Processing* **81**(7), 1513-1523.

[21] C. Braccini, F. Cocurullo, F. Lavagetto 1995 A fast algorithm for high quality vector quantization codebook design. *Lecture Notes in Computer Science*, 974, 643-648.

[22] Q. Chen, J. Yang, J. Gou 2005 Image compression method using improved PSO vector quantization. Lecture Notes in Computer Science, 3612, 490-495.

[23] C.M. Ou 2010 Vector quantization based on steady-state memetic algorithm. *Journal of Marine Science and Technology* **18**(4):553-557.

Authors

Zhao Mengling, <1974.07>,xi'an, shanxi, P.R. China

Current position, grades: the Ph.D. , Vice Professor of science of Xi’an University of Science and Technology, China.

University studies: I am one of the members of school of science of Xi’an University of Science and Technology. From 1994 to 1998, I studied in the Mathematics Department of North-West University, majoring in applied mathematics. In July 1998, I successfully graduated with the bachelor degree.

Scientific interest: His research interest fields include Optimization algorithm.

Publications: more than 6 papers published in various journals.

Experience: He has teaching experience of 16 years, has completed three scientific research projects.

Liu Hongwei , 1967.02, xi'an, shanxi, P.R. China

Current position, grades: Ph.D., the Professor of Xidian University in Xi’an PR China.

University studies: I am an advisor of PH.D.candidates, I earned my doctorate in Applied mathematics from School of Mathematics and Statistics, Xidian University in Xi’an PR China.

Scientific interest: His research interest fields include optimization theory and algorithm research，cone planning theory and algorithm research, mathematical methods in data mining，variational inequalities and complementarity problems，the optimization problem in machine learning research.

Publications: more than 8 papers published in various journals.

Experience: He has teaching experience of 20 years, has completed three scientific research projects.