Short Paper

A Tabu Search Based Maximum Descent Algorithm for VQ Codebook Design

HSIANG-CHEH HUANG, SHU-CHUAN CHU[*], JENG-SHYANG PAN[**]
AND ZHE-MING LU[***]

Department of Electronics Engineering
National Chiao Tung University
Hsinchu, 300 Taiwan
[*]*School of Informatics and Engineering*
Flinders University of South Australia
Australia
[**]*Department of Electronic Engineering*
National Kaohsiung University of Applied Sciences
Kaohsiung, 807 Taiwan
E-mail: jspan@cc.kuas.edu.tw
[***]*Department of Automatic Test and Control*
Harbin Institute of Technology
Harbin, China

A maximum descent (MD) method has been proposed for vector quantization (VQ) codebook design. Compared with the traditional generalized Lloyd algorithm (GLA), the MD algorithm achieves better codebook performance with far less computation time. However, searching for the optimal partitioning hyperplane of a multidimensional cluster is a difficult problem in the MD algorithm. Three partition techniques have been proposed for the MD method in the literature. In this paper, a new partition technique based on the tabu search (TS) approach is presented for the MD algorithm. Experimental results show that the tabu search based MD algorithm can produce a better codebook than can the conventional MD algorithms.

Keywords: VQ, codebook design, maximum descent method, tabu search, GLA

1. INTRODUCTION

Vector quantization (VQ) has been extensively and successfully used in speech encoding and image compression [1]. The k-dimensional, N-level vector quantizer is defined as a mapping from a k-dimensional Euclidean space R^k into a certain finite set C of R^k. This finite set $C = \{Y_1, Y_2, \ldots, Y_N\}$ is called a VQ codebook. Each $Y_i \in R^k$ in codebook C is called a codeword, $i = 1, 2, \ldots, N$. The VQ quantizer consists of two procedures: an encoder and a decoder. The encoder assigns each input vector X to an index i, which

Received November 24, 1999; revised April 17, 2000; accepted July 19, 2000.
Communicated by Yung-Nien Sun.

points to the closest codeword Y_i in the codebook. The decoder uses the index i to look up the codeword Y_i in the codebook. The squared Euclidean distortion measure is often used to measure the distortion between the input vector $X = \{x_1, x_2, \ldots, x_k\}$ and the codeword $Y_i = \{y_{i1}, y_{i2}, \ldots, y_{ik}\}$, i.e.,

$$d(X, Y_i) = \sum_{l=1}^{k} (x_l - y_{il})^2 . \tag{1}$$

The key problem with VQ is to generate a good codebook from a number of training vectors. The codebook design problem is essentially a clustering problem where a training set $S = \{X_1, X_2, \ldots, X_M\}$ is clustered into N subsets $S_1, S_2, \ldots, S_N$, which satisfy

$$\bigcup_{i=1}^{N} S_i = S \tag{2}$$

and

$$S_i \bigcap S_j = \Phi \text{ if } i \neq j . \tag{3}$$

The centriod of the subset S_i is the codeword Y_i, and if

$$d(X_l, Y_i) = \min_{1 \le j \le N} d(X_l, Y_j) , \tag{4}$$

then $X_l \in S_i$. The aim of the codebook design algorithm is to minimize the total distortion defined as follow:

$$D = \sum_{l=1}^{M} \min_{1 \le j \le N} d(X_l, Y_j) \cdot \tag{5}$$

A generalized Lloyd clustering algorithm [2] proposed by Linde, Buzo, and Gray, called the LBG algorithm or GLA, is typically used to generate VQ codebooks. However, this iterative algorithm depends on the initial codebook, often obtains the local optimal codebook and requires intensive computation. Thus, the two prominent problems in codebook design are how to approach the global optimum and how to reduce the computational complexity. In order to obtain better codebooks, Vaisey and Gersho [3] combined the simulated annealing (SA) technique and the LBG iteration in VQ codebook design. In their SA-GLA algorithm, they took the codebook design as a combinatorial problem by perturbing the encoder of a vector quantizer in order to let the LBG algorithm converge again and again so as to obtain lower overall distortion. This method can improve codebook performance; however, the computational time used by this algorithm is at least several times that used by the LBG algorithm. The stochastic relaxation (SR) approach [4] has also been proposed to improve codebook performance. The basic idea of the SR approach is to perturb the solution by adding some values to the codewords during each iteration. Many methods designed to reduce the computational time have appeared in the literature. The subspace distortion method [5] attempts to reduce the computation burden by reducing the dimension of the distortion measure in the LBG algorithm. The pairwise nearest neighbor (PNN) algorithm [6] is a bottom-up clustering algorithm that generates a codebook by merging nearest training vector clusters until the desired number of codewords is obtained. The codebooks generated by both methods are not as good as the codebook generated by the LBG algorithm although the computational time is reduced by several times.

Designed to generate better codebooks while reducing the computational time, the maximum descent (MD) algorithm [7] has been proposed for VQ codebook design. This algorithm begins by treating the training vector set $S = \{X_1, X_2, \ldots, X_M\}$ as a global cluster. Then, the algorithm generates the required number of clusters one by one, subject to the maximum distortion reduction criterion, until the desired number of codewords is obtained. Compared with the LBG algorithm, the codebook performance is improved, and the computational time is substantially reduced. However, searching for the optimal partitioning hyperplane in a multidimensional cluster is a difficult problem with the MD algorithm. Three techniques [8] have been presented for searching for the optimal partitioning hyperplane with the MD method. However, these techniques all restrict the searching range to hyperplanes that are perpendicular to the basis vectors of the vector space that can be obtained using discrete cosine transform (DCT), so it is difficult for them to find the global optimal partitioning hyperplane. In order to find the global or nearly global optimal partitioning hyperplane, the tabu search (TS) approach first proposed by Glover [9] is introduced in this paper. The tabu search approach is a global optimization technique with short-term memory, that can be used to solve many difficult combinatorial optimization problems. In previous works [10-11], the tabu search approach has been successfully applied to codebook index assignment over channel noise and texture segmentation. The main problem with MD is determining how to split one cluster into two clusters with minimal mean squared error. This is a typical combinatorial optimization problem, so the tabu search approach is suitable for solving it.

Section 2 presents the main idea of the maximum descent algorithm and three conventional methods used to search for the optimal partitioning hyperplane. Section 3 presents the proposed tabu search based maximum descent algorithm. Section 4 describes experiments carried out to compare the codebook performance of the proposed algorithm with that of the LBG algorithm, the simulated annealing method and the basic MD algorithms with the three partition techniques presented in [8]. Section 5 concludes the paper.

2. THE MAXIMUM DESCENT ALGORITHM

Let us consider the design of an N-level codebook $C = \{Y_1, Y_2, \ldots, Y_N\}$ for training set $S = \{X_1, X_2, \ldots, X_M\}$, where $Y_i \in R^k$, $i = 1, 2, \ldots, N$, $X_m \in R^k$, $m = 1, 2, \ldots, M$, $M >> N$. The maximum descent algorithm first views the training vector set S as a global cluster. This cluster is split into two new clusters by an optimal partitioning hyperplane. One of these two clusters is then further partitioned into two clusters so as to generate three new clusters based on a maximum distortion reduction criterion. For the general case of $L(L>2)$ clusters, the MD method attempts to generate L+1 clusters by splitting one of the former L clusters into two new clusters and keeping the other L-1 clusters unchanged such that the distortion can be reduced as much as possible. This procedure is performed until the desired number of clusters is obtained. Finally, the centriods of these clusters are taken as codewords.

Now, we will to investigate the distortion reduction resulting from splitting a cluster into two new clusters. Without loss of generality, we consider the case in which the original training set S has been partitioned into L clusters, i.e., $S = \{S_1, S_2, \ldots, S_L\}$. Assume that a hyperplane $H_i(U, v) = \{Z \in R^k : U^T Z = v\}$ further partitions the cluster S_i into two non-empty clusters given by

$$S_{ia} = \{Z \in S_i : U^T Z < v\},$$
$$S_{ib} = \{Z \in S_i : U^T Z \geq v\}, \tag{6}$$

where $U \in R^k$, $v \in R$, $1 \leq i \leq L$, and T denotes the transpose.

The centroid of cluster S_i can be defined as

$$Y_i = \frac{\sum_{j:X_j \in S_i} X_j}{n_i}, \tag{7}$$

where n_i is the number of input vectors that belong to cluster S_i. When the centriod Y_i is used to quantize all the vectors inside S_i , the induced total squared Euclidean distortion $D(S_i)$ is

$$D(S_i) = \sum_{j:X_j \in S_i} \left\| X_j - Y_i \right\|^2, \tag{8}$$

where $\|X_j - Y_i\|^2$ denotes the squared Euclidean distortion between X_j and Y_i. Then the reduction in distortion induced by partitioning the cluster S_i into two clusters using the partition plane H_i can be expressed as

$$r_i = D(S_i) - [D(S_{ia}) + D(S_{ib})]. \tag{9}$$

Direct computation of Eq. (9) requires many calculations of squared distances. In order to reduce the computational complexity of Eq. (9), we can easily rewrite Eq. (9) as follows (the proof for this can be found in [8]):

$$r_i = n_i \cdot \frac{n_{ia}}{n_{ib}} \cdot \left\| Y_i - Y_{ia} \right\|^2 \tag{10}$$

or

$$r_i = n_i \cdot \frac{n_{ib}}{n_{ia}} \cdot \left\| Y_i - Y_{ib} \right\|^2, \tag{11}$$

where n_i, n_{ia} and n_{ib} are the number of vectors in S_i, S_{ia} and S_{ib}, respectively. Y_i, Y_{ia} and Y_{ib} are the centroids of clusters S_i, S_{ia} and S_{ib}, respectively.

From equation 10 we can easily prove that r_i is always greater than zero, i.e., that the reduction in distortion is never less than zero. In order to obtain the maximum reduction function $\hat{r}_i$ for cluster S_i, we need to use some techniques to search for the optimal partitioning hyperplane $\hat{H}_i$ for cluster S_i. To form L+1 clusters from the L clusters with maximum distortion reduction, the maximum distortion reduction functions of all the clusters need to be calculated (Fig. 1a); then, one of the L clusters S_p which satisfies $\max_{1 \leq i \leq L}\{\hat{r}_i\} = \hat{r}_p$ is split into two new clusters while the remaining L-1 clusters stay unchanged. In the step that obtains L+2 clusters from the L+1 clusters, only two maximum distortion reduction functions, i.e., $\hat{r}_{pa}$ and $\hat{r}_{pb}$, of the newly formed clusters S_{pa} and S_{pb} (Fig. 1b) need to be computed since the $\hat{r}_i$ s of all other clusters have already been

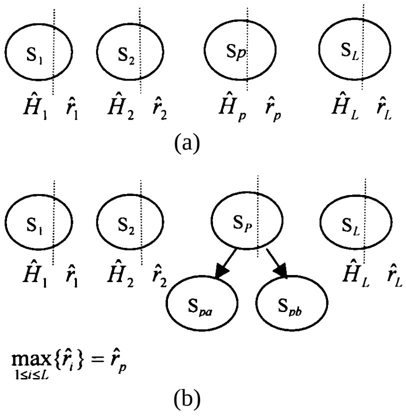

Fig. 1. The optimal formation of (L+1) clusters that provides maximum reduction of overall distortion.

computed in the previous step. Thus, in each step in forming an additional cluster, only two optimal partitioning hyperplanes need to be found. Based on this maximum descent criterion, the clusters are split one by one until the required N clusters are obtained. We can easily prove that only 2N-3 optimal partition searches are required to design an N-level vector quantizer, so the number of optimal partition searches is proportional to the size of the codebook.

It is noted that the MD algorithm has an advantage in that it tends to partition clusters that are densely populated and never generates a cluster that includes only one vector, which sometimes happens with the conventional LBG algorithm. However, exhaustive searching for the optimal partitioning hyperplane of a multidimensional cluster is a difficult and computationally intensive problem, and it is very hard to carry out in practice. In order to reduce the complexity, three methods of searching for the optimal partitioning hyperplane were presented in [8] and are geiven below.

2.1 Constrained Exhaustive Search

This method searches for the optimal hyperplane among the hyperplanes that are perpendicular to the basis vectors of the vector space that can be obtained by means of the discrete cosine transform of the input image. For a cluster S_i, the lower and upper bounds of the vectors on one of the DCT basis axes are found. The algorithm first assigns a splitting threshold T to the lower bound and places the training vectors whose projections on this axis are smaller than T into cluster S_{ia} and puts the remaining ones into S_{ib}. r_i for threshold T is computed. The threshold T is then increased by means of a predetermined incremental step. The procedures are repeated until T reaches the upper bound. The optimal partitioning hyperplane and the corresponding $\hat{r}_i$ on this axis are recorded. Based on the same procedure, all other optimal partitioning hyperplanes for other basis axes are found. By finding the maximum $\hat{r}_i$ among all the $\hat{r}_i$ s obtained, the constrained global optimal partitioning hyperplane and the corresponding distortion reduction function are obtained for cluster S_i.

2.2 Successive Search

Constrained exhaustive search is a computationally intensive technique. Its complexity can be reduced by using a successive search method based on the idea that the maximum reduction can often be obtained when $D(S_{ia})$ is equal to $D(S_{ib})$. This method starts by assigning a threshold T half way between the upper and lower bounds on an axis. Then $D(S_{ia})$ and $D(S_{ib})$ are evaluated. If $D(S_{ia}) > D(S_{ib})$, the threshold is set to T_1, which is half way between T and the lower bound. If $D(S_{ia}) \leq D(S_{ib})$, T_1 is set to a value half way between T and the upper bound. $D(S_{ia})$ and $D(S_{ib})$ are computed for T_1, and the next threshold T_2 can be determined using the same method described above. This procedure is repeated until the difference between $D(S_{ia})$ and $D(S_{ib})$ is smaller than a pre-defined value. The optimal hyperplane and the distortion reduction value are thus obtained. Similarly, successive search needs to be performed over all the basis axes in order to find the maximum distortion reduction value and the optimal hyperplane.

2.3 Fast LBG Search

If the optimal partition hyperplane is restricted to being perpendicular to only one of the basis vectors of the transformed vector space, then we can use the following fast LBG algorithm to perform binary splitting of a cluster S_i into S_{ia} and S_{ib} so as to reduce the amount of computation. In this method, the difference in the Euclidean distances $d(X, Y_{ia}, Y_{ib})$ between the input vector $X \in S_i$ and the two centroids Y_{ia} and Y_{ib} is defined as follows:

$$\begin{aligned} d(X, Y_{ia}, Y_{ib}) &= \sum_{j=1}^{k}(x_j - a_j)^2 - \sum_{j=1}^{k}(x_j - b_j)^2 \\ &= -2\sum_{j=1}^{k}(a_j - b_j)x_j + \sum_{j=1}^{k}(a_j{}^2 - b_j{}^2), \end{aligned} \tag{12}$$

where $X = (x_1, x_2, \ldots, x_k)$, $Y_{ia} = (a_1, a_2, \ldots, a_k)$ and $Y_{ib} = (b_1, b_2, \ldots, b_k)$.
Based on Eq. (12), if

$$\sum_{j=1}^{k}(a_j - b_j)x_j > \frac{1}{2}\sum_{j=1}^{k}(a_j{}^2 - b_j{}^2), \tag{13}$$

then the vector X is closer to the cluster S_{ia}, so the vector X can be placed in S_{ia}. Otherwise, it is placed in S_{ib}. Since $(a_j - b_j)$ and $\sum_{j=1}^{k}(a_j{}^2 - b_j{}^2)$ do not change throughout the partitioning process and can be pre-calculated, the amount of computation is greatly reduced.

3. TABU SEARCH BASED MD ALGORITHM

The above three methods restrict the searching range to hyperplanes that are perpendicular to the basis vectors of the vector space that can be obtained by means of discrete cosine transform of the input image, so it is difficult for them to find the global optimal hyperplane. In this section, we will introduce the tabu search technique which can be used to search for the optimal hyperplane.

The basic idea of tabu search is to explore the search space of all feasible solutions by means of a series of moves and to forbid some search directions at the present iteration in order to avoid cycling and jump off local optima. The elements of moving from the current solution to its selected neighbor are partially or completely recorded in the tabu list for the purpose of forbidding reversal of replacement in a number of future iterations. The tabu search approach begins with test solutions generated randomly and evaluates the objective function for these solutions. If the best of these solutions is not tabu or if it is tabu but satisfies the aspiration criterion, then this solution is selected as the new current solution to be used to generate test solutions for the next iteration. It is called the aspiration criterion if the test solution is a tabu solution but the objective value is better than the best value of all the iterations.

The basic task of the MD algorithm is to find the optimal partitioning of a certain cluster S into two clusters S_a and S_b which maximizes the following objective function

$$r = n \cdot \frac{n_a}{n_b} \cdot \left\| Y - Y_a \right\|^2 \tag{14}$$

or

$$r = n \cdot \frac{n_b}{n_a} \cdot \left\| Y - Y_b \right\|^2. \tag{15}$$

Here n, n_a and n_b are the number of vectors in S, S_a and S_b respectively, where $n = n_a + n_b$. Y, Y_a and Y_b are the centroids of clusters S, S_a, and S_b, respectively.

In order to describe the proposed algorithm, the corresponding indices of the training vectors that belong to cluster S are used to form a solution. Every solution can be divided into two parts. The corresponding training vectors in the first part belong to cluster S_a, and the corresponding training vectors in the second part belong to cluster S_b. Thus the optimal partitioning of S into S_a and S_b can be found using the following algorithm.

Let P_t, p_c and p_b be the test solutions, the best solution of the current iteration and the best solution of all the iterations, respectively, where $P_t = \{p_1, p_2, \ldots, p_{Ns}\}$, $p_i = \{p_{ia}, p_{ib}\}$ is one of test solutions, p_{ia} includes the indices of the training vectors that belong to S_a, p_{ia} includes the indices of the training vectors that belong to S_b, $1 \le i \le N_S$, and N_S is the number of test solutions, $p_c = \{p_{ca}, p_{cb}\}$ and $p_b = \{p_{ba}, p_{bb}\}$.

Let R_t, r_c and r_b denote the objective function values for the test solutions, the objective function value for the best solution of the current iteration, and the objective function value for the best solution of all the iterations, respectively, where $R_t = \{r_1, r_2, \ldots, r_{Ns}\}$, r_i is the objective function of p_i, $1 \le i \le N_S$, and N_S is the number of test solutions. The initail test solutions are generated randomly. After the first iteration, the test solutions are generated from the best solution of the current iteration by means of some moves. The tabu list memory stores the moved indices only. It is a tabu condition if the moved indices used to generate the test solution from the best solution of the current iteration are the same as any records in the tabu list memory. The algorithm is given as follows:

Step 1. Set the tabu list size be T_S, the number of test solutions be N_S and the maximum number of iterations be I_m. Set the iteration counter $i = 1$, and let the insertion point of the tabu list $t_l = 1$. Generate N_S initail solutions $P_t = \{p_1, p_2, \ldots, p_{Ns}\}$

randomly, calculate the corresponding objective values $R_t = \{r_1, r_2, \ldots, r_{Ns}\}$ according to Eq. (14) and find the current best solution $p_c = p_j$, $j = \arg\max_l r_l$, $1 \le i \le N_S$. Set $p_b = p_c$ and $r_b = r_c$.

Step 2. Copy the current best solution p_c to each test solution p_i, $1 \le i \le N_S$. For each test solution $p_i = \{p_{ia}, p_{ib}\}$, $1 \le i \le N_S$, generate a new solution by means of the following three substeps.

Step 2.1. If $n_{ia} \ge 2$ and $n_{ib} \ge 2$, select p_{ia} or p_{ib} randomly; else if $n_{ia} = 1$ and $n_{ib} \ge 2$, select p_{ib}; else if $n_{ib} = 1$ and $n_{ia} \ge 2$, select p_{ia}; else if $n_{ia} = 1$ and $n_{ib} = 1$, select neither of them.

Step 2.2. If p_{ia} is selected, then select an index in p_{ia} randomly and move it to p_{ib}; else if p_{ib} is selected, then select an index in p_{ib} randomly and move it to p_{ia}. If neither of them is selected, the solution remains unchanged.

Step 2.3. Calculate the corresponding objective value r_i for the new test solution.

Step 3. Sort r_1, r_2 ,…, r_{Ns} in decreasing order. From the best new test solution to the worst new test solution,if the new test solution is a non-tabu solution or if it is a tabu solution but its objective value is better than the best value of all iterations r_b, then choose this new solution as the current best solution p_c, choose its objective value as the current best objective value r_c, and go to step 4; otherwise, try the next new test solution. If all new test solutions are tabu solutions, then go to step 2.

Step 4. If $r_b < r_c$, set $p_b = p_c$ and $r_b = r_c$. Insert the moved index of the current best solution p_c into the tabu memory list. Set the insertion point of the tabu list $t_l = t_l + 1$. If $t_l > T_S$, set $t_l = 1$. If $i < I_m$, set $i = i + 1$ and go to step 2; otherwise, record the best solution and terminate the algorithm.

4. PERFORMANCE COMPARISONS

In order to demonstrate the efficiency of the proposed algorithm, the LBG algorithm, the SA algorithm, the conventional MD algorithms and the tabu search based MD algorithm were all implemented to generate codebooks. For the SA algorithm, the initial temperature T_0 was 35, and the temperature was decreased by 1% after each iteration step until the number of iterations reached 30. Because each iteration of SA included the LBG algorithm, the CPU time of the SA was about 25 times that of the LBG algorithm in the experiments. Here, the parameters of tabu search were $T_S = 20$, $N_S = 20$, and $I_m = 200$. All the tests were run on a Pentium II PC running at 233MHz. Three images, LENA, PEPPERS and F-16, with a resolution of 512 × 512 pixels, 8bits/pixel, were used. The image LENA was used to generate codebooks of size 256 with dimension 16(4 × 4), and the images PEPPERS and F-16 were used to test the coding performance of the codebooks. Results are shown in Tables 1 and 2. Table 1 compares the six algorithms based on CPU time and PSNR when generating codebooks of size 256 were generated. From Table 1, we can see that the PSNR of the tabu search based MD algorithm was improved by 0.5dB compared with the constrained exhaustive MD algorithm and by 1.4dB compared with the LBG algorithm although it needed more CPU time compared with the constrained exhaustive MD algorithm. For the images outside the training set, Table 2 shows the performance of codebooks of size 256 that were generated by different algorithms. From Table 2, we can see that the PSNR of the tabu search based MD algorithm was

Table 1. Performance comparison of LBG, SA and various MD algorithms for the image within the training set.

Algorithm	CPU time (Sec)	PSNR(dB)
LBG	360.5	30.23
SA	9035.2	31.48
Constrained Exhaustive Search MD	120.2	31.14
Successive Search MD	15.3	30.96
Fast LBG Search MD	3.8	31.08
Tabu Search Based MD	1023.1	31.64

Table 2. Performance comparison of LBG, SA and various MD algorithms for images outside the training set.

Algorithm	PSNR of the coded images (dB)	
	Peppers	F-16
LBG	28.45	26.37
SA	29.01	26.93
Constrained Exhaustive Search MD	28.94	26.82
Successive Search MD	28.85	26.75
Fast LBG Search MD	28.87	26.79
Tabu Search Based MD	29.14	27.08

improved by 0.2dB compared with the constrained exhaustive MD algorithm and by 0.7dB compared with the LBG algorithm. Although the proposed algorithm needs more CPU time than the conventional MD algorithms, it needs less time than the SA algorithm and has better performance than the SA algorithm and the other algorithms. In conclusion, the proposed algorithm can obtain better reconstructed image quality than the LBG algorithm and all the conventional MD methods not only for an image in the training set, but also images outside the training set.

5. CONCLUSIONS

In this paper, a tabu search based maximum descent method for generating codebooks for vector quantization has been proposed. This method generates better codebooks than does the LBG algorithm or the conventional MD algorithms although it needs more CPU time than the LBG algorithm and the conventional MD algorithms. Not only is the performance improved, but the computation time is also reduced based on a comparison of the proposed algorithm with the simulated annealing method. Since codebook design is performed off line for most applications, the proposed tabu search based maximum descent algorithm can be viewed as a good approach to codebook design.

REFERENCES

1. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Boston, Kluwer Academic Publishers, 1992.

2. Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design,” *IEEE Transactions on Communications*, Vol. COM-28, 1980, pp. 84-95.
3. J. Vaisey and A. Gersho, “Simulated annealing and codebook design,” in *Proceedings of International Conference on Acoustics, Speech and Signal Processing’88*, 1988, pp. 1176-1179.
4. K. A. Zeger and A. Gersho, “Stochastic relaxation algorithm for improved vector quantizer design,” *Electronic Letters*, Vol. 25, No. 14, 1989, pp. 896-898.
5. L. M. Po, “Novel subspace distortion measurement for efficient implementation of image vector quantizer,” *Electronic Letters*, Vol. 26, No. 29, 1990, pp. 480-482.
6. W. H. Equitz, “A new vector quantization clustering algorithm,” *IEEE Transactions on ASSP*, Vol. 37, No. 10, 1989, pp. 1568-1575.
7. C. K. Ma and C. K. Chan, “Maximum descent method for image vector quantization,” *Electronics Letters*, Vol. 27, No. 12, 1991, pp. 1772-1773.
8. C. K. Ma and C. K. Chan, “A fast method of designing better codebooks for image vector quantization,” *IEEE Transactions on Communications*, Vol. 40, No. (2/3/4), 1994, pp. 237-242.
9. F. Glover and M. Laguna, *Tabu Search*, Boston, Kluwer Academic Publishers, 1997.
10. J. S. Pan and S. C. Chu, “Non-redundant VQ channel coding using tabu search strategy,” *Electronics Letters*, Vol. 32, No. 17, 1996, pp. 1545-1546.
11. J. S. Pan, J. W. Wang, H. L. Fang, and C. H. Chen, “A modified tabu search approach for texture segmentation using 2-D non-separable wavelet frames”, *The 10th IEEE International Conference on Tools with Artificial Intelligence*, pp. 474-481.

H. C. Huang (黃祥哲) got the Ph.D. in Electronics Engineering from National Chiao Tung University in 2001. Currently he is the research associate at Department of Electronics Engineering, National Chiao Tung University. His current research interests include pattern recognition, image compression and watermarking.

S. C. Chu (朱淑娟) received the B. M. degree in Industrial Engineering and Management from the National Taiwan Institute of Technology, Taiwan, in 1988. She is currently a Ph.D. candidate at the Flinders University of South Australia, Adelaide, Australia. Her current research interests include data mining and clustering.

J. S. Pan (潘正祥) received the B. S. degree in Electronic Engineering from the National Taiwan Institute of Technology, Taiwan, in 1986, the M. S. degree in Communication Engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in Electronic Engineering from the University of Edinburgh, U.K., in 1996. Currently, he is a Professor in the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan. His current research interests include pattern recognition, speech coding and image processing.

Z. M. Lu (陸哲明) received the B. S. degree, M. S. degree and Ph.D. all in Electronic Engineering from the Harbin Institute of Technology, Harbin, China, in 1995, 1997 and 2001, respectively. He is currently an Associate Professor at the Harbin Institute of Technology. His current research interests include speech coding and image processing.