Vector Quantizer Design by Conjugate Gradient Optimized Hyperplane

Kam-Tim Woo, Kam-Fai Chan and Chi-Wah Kok

Dept. EEE, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, HONG KONG

Abstract

An Vector Quantizer design method by adpative hyperplane geneartion using conjugate gradient optimization is proposed. The generated hyperplane is a perpendicular bisector of the clustering set centroids at each stage of the K-dimensional search tree, thus eliminated misclassification error associated with hyperplane based vector quantization. Simulation results on Vector quantization image coding is presented and compared with that obtained by other algorithms in literature. Where the results showed that the proposed algorithm can achieve better PSNR image coding results than that obtained by other algorithms. The generated K-dimensional search tree vector quantizer facilities computational efficient quantization process.

1 Introduction

Vector quantization is a very popular lossy compression tool applied in speech and image codings. The performance of vector quanitzer heavily depends on the codebook used, and the search method applied in the quantization process. A lot of efforts have been placed in searching for better codebook generation method and reducing the computational complexity to search the optimal code vector in vector quantization process.

LBG algorithm [1] is the most commonly used vector quantization codebook design method. This method requires a large sample data vectors for training and generates a locally optimal codebook by a K-means alike iteration which takes a long time to converge. Furthermore, the performance of the generated codebook heavily depends on the initial condition used in training. Other codebook designed algorithms partition the space occupied by the sample vectors space into subspaces and organize them to facilite an efficient search method, such as K-dimensional (K-d) search tree. Hyperplane (linear discrimination function) is a frequently used partitioning method in K-d search tree VQ algorithms [3, 4, 5, 7]. Hyperplane splitting algorithm is a kind of Tree-structured VQ [1] where the search tree growth by splitting the cluster set associated with the terminal nodes in the search tree to form new branches with terminal nodes containing the partitioned clusters and hence the codewords generated after partition. The splitting process continues until the number of partitioned clusters (i.e. terminal nodes) equals to the desired codebook size.

To further lower the search complexity of the K-d tree VQ, [4] proposed to choose hyperplane that are perpendicular to the base axis and partition the subspace with maximum variance. The hyperplane in [4] is constrained to pass through the centroid of the data set in the partitioned subspace. Noticed that the performance of the K-d tree VQ depends on choosing a suitable subspace for subdivision. Subdividing subspace of data set that has maximum variance does not necessary results in good clustering [4]. In [5], a maximum reduction in sum of square error (SSE) hyperplane partition criteria is proposed for orthogonal hyperplane. The restrictions of orthogonal hyperplane splitting is lifted in [3], and demonstrated that arbitrary hyperplane can achieve bigger SSE reduction than that of [4], which results in better generated codebook. Further improvement can be obtained by lifting the constraint to restrict the hyperplane to pass through the centroid of the data set to be partitioned. An adaptive RWM cut [6] approach suggested an adaptive algorithm to select the position of the hyperplane.

To design better vector quantizer, it should be noticed that there are two main criteria for obtaining optimal quantizer. The first is the decision boundaries should be located at the mid-points between the reconstruction levels of neighboring clusters. The second condition is the quantized vectors should be the centroids of quantized clusters. The second condition can be easily satisfied and has been adopted in almost all the VQ codebook generation algorithms. In order to obtain a hyperplane which satisfies the first condition, [7] proposed an iterative algorithm to select the best position for the hyperplane to achieve maximum reduction in SSE while located at the mid-point of the partitioned cluster sets. Simulation results have shown that better codebook can be achieved. However, the constructed K-d search tree in [7] still suffers from hyperplane related misclassification due to the fact that the hyperplane the decision boundary is not equal distance to the quantized vectors of the partitioned cluster sets. The first criteria for obtaining optimal quantizer can be satisfied when the hyperplane passes through the mid-point of the quantized vectors of the partitioned clusters, and perpendicular to the line that joins the two quantized vectors. The hyperplane obtained from [7] may not satisfy this criteria. As a result, the associated misclassification problem not only increase the computational complexity of the vector quantizer, but also limited the performance of the designed codebook. This paper proposed a hyperplane based vector quantizer design algorithm, such that the obtained hyperplane will bisects the cluster space in half-space sense and thus satify the first criteria of the optimal quantizer. An iterative algorithm is derived to generate "adaptive arbitrary hyperplane" that select the best position and orientation to partition the cluster set and achieve maximum reduction in SSE. Furthermore, the generated hyperplane is a perpendicular bisector of the centroids of the partitioned cluster sets. Vector quantization image coding results are presented which showed that the proposed algorithm can achieve better PSNR than that of [1, 3, 6, 7] for both images inside and outside the training sets.

2 The proposed algorithm

The search for optimal partitioning hyperplane in multi-dimensional clustering is a difficult and computational intensive problem. There are two fundamental issues associated with the hyperplane generation problem, which are i) selecting an appropriate subspace for subdivision, and ii) selecting a suitable hyperplane to subdivide the chosen data cluster. Similar to [4], the proposed algorithm chooses to partition the data cluster that results in maximum reduction in SSE. As a result, the first problem cannot be solved without solving the second problem first, i.e. choosing a suitable hyperplane to subdivide the data cluster. Following [6], the hyperplane are chosen to be perpendicular to the principle component of the input process. Let X be the input signal vectors with dimension $1 \times N$. The zero mean covariance matrix R of X is given by

$$R = E[(X - \overline{X})^T (X - \overline{X})], \tag{1}$$

where E is the expectation operator, $\bar{X}$ is the mean of the input signal vectors, and T denotes transpose. The principle component of the covariance matrix R is given by the eigenvector a associated with the largest absolute eigenvalue. The principle component of the input process indicates the vector direction that obtain the maximum projected variance from the input data. As a result, choosing a hyperplane perpendicular to the principle component of the input signal should achieve maximum reduction in SSE. The splitting is performed according to the following rules.

$$S_1 = \{X \in S | (X - C)a^T > 0\} \tag{2}$$

$$S_2 = \{X \in S | (X - C)a^T \leq 0\} \tag{3}$$

where S is the data set to be splitted, and C is the position of the hyperplane.

The centroids of the partitioned clusters are chosen as the quantized vectors. As a result, the second criteria for optimal quantization as discussed in Section 1 is satisfied. Meanwhile, the first condition can be satisfied by hyperplane that is the perpendicular bisector of the centroids of the partitioned cluster sets. An adaptive algorithm is proposed which updates the location and orientation of the hyperplane through two nested iterative loops, such that the final hyperplane will satisfy the first condition of optimal quantization. In *Iteration 1*, the position of the hyperplane is updated using the algorithm presented in [7]. In each iteration, the position of the hyperplane is updated to the mid-point of the centroids of cluster S_1 and S_2. S_1 and S_2 are updated according to eq.(2,3) with the new hyperplane. In *Iteration 2*, the orientation of the hyperplane is updated to be perpendicular to the line joining the centroids of S_1 and S_2. S_1 and S_2 are updated according to eq.(2,3) with the new hyperplane. The two iterations are nested and repeated until converge, such that the difference between the location, and hyperplane orientation obtained in the current loop and that of the last loop is smaller than a per-determined ϵ and ζ respectively.

2.1 Conjugate Gradient Optimization

The discussed iterations in previous section is a kind of linear-search algorithm. It is very simple to implement, however, it has a aignificant disadvantage that it may take an infinite iterations for convergence. In order to solve this problem, we proposed a conjugate gradient algorithm for the search of optimal hyperplane orientation. The orientation of the hyperplane, as specified by the normal vector a is updated with a portion, r, of one of the orthogonal basis. In general, there are $2N$ possible candidates of direction for rotating a hyperplane plane. The candidate vector that form the smallest angle with b, the vector passing through the two centroids of S_1 and S_2, is chosen to be the update vector. Note that, to achieve good convergence, the conjugate gradient algorithm will not chose the same orthogonal axis is consecutive iterations.

2.2 Multiple Initial Position Candidates

Simulation results have showed that the performance and number of iterations in the proposed hyperplane generation method is sensitive to the initial hyerplane location. Similar to [6], multiple candidates are applied, such that the centroid $O = \frac{1}{n}\sum_{i=1}^{n} X_i$, radius square weighted mean (RSWM) $P = \frac{1}{W}\sum_{i=1}^{n} \|X_i - \overline{X_i}\|^2 X_i$; $W = \sum_i^n w_i^2$ where i denotes the i-th element of the vector signal. and median $Q = (P + \overline{X})/2$ are considered as the initial guesses.

We proposed to let the initial hyperplane passes through the candidate with the smallest SSE after partition, while the orientation of the initial hyperplane is chosen to be perpendicular to the eigenvector R of eq.(1) that associates with the largest absolute eigenvalue. Simulation results showed that the above initial guess will lead to better codebook than that obtained by hyperplane passing though the centroid. It is also observed that the above algorithm convergence faster.

2.3 Construction of K-d Search Tree

A K-d Bentley tree that allows multidimensional search using the above hyperplane generation method can be constructed [4]. The input data set is first splitted into two data clusters by the described hyperplane. The cluster set that results in maximum reduction in SSE after splitting is selected for further splitting by the above hyerplane generation method. This splitting procedure is repeated until N-disjoint cluster sets are obtained. The centroids of each cluster set are the codevectors of the generated VQ codebook.

1. Compute the principle component vector of the input signal covariance matrix R as defined in eq.(1).
2. Compute the centroid O, RSWM P and median Q of the data set.
3. Partition the data set by hyperplane perpendicular to the principle component vector in step 1 and passes through either O, P and Q, such that minimum SSE is achieved through the partition.
4. Compute the centroids of the partitioned data set.
5. Update the position of hyperplane as the mid-point of the two centroids from step 4.
6. Compute vector b which passes through the two centriods of two cluster sets S_1 and S_2.
7. Define a $1 \times K$ vector $c_i = [0, 0, ..., 0, 1, 0, ...]$ with every column equal to zero except the i-th column which equals to 1 for all i.
8. Compute the updating vectors as $a + r < (b - a), c_i >$ and $a + r < (-b - a), c_i >$ for all i, where r and $< \cdot, \cdot >$ are the step size in each update process and dot product operator respectively.
9. The updated vector that achieve the smallest angle with b is used to update the orientation of the hyperplane and record the selected axis k.
10. If same axis is selected in the last iteration, the updated vector that achieve the second smallest angle with b is used to update the orientation of the hyperplane.
11. Computes the new centroids and normal vector of the paritioned data set.
12. If the difference between updated position and normal vector and that from last iteration is smaller than ϵ and ζ respectively, goto step 14.
13. Update the partitioned data set according to eq.(2,3). Goto step 4.
14. Compute the SSE of the partitioned data sets.
15. If the number of disjoint data sets equals to the desired codebook size, then stop.
16. Select the data sets with the minimum SSE and goto step 1.

2.4 Step Size Adaptation

Further improvement can be obtained by adapting the step size r. We proposed to adapt the step size according to the rate of change in total SSE of the splitting data. The step size r at the $k+1$-th iteration should proportional to

$$r_{k+1} \propto \frac{SSE_4(k) - SSE_4(k-1)}{SSE_4(k-1) - SSE_4(k-2)}, \tag{4}$$

where $SSE_4(k)$ is the total SSE in each splitting of data set in k^{th} iteration. When $SSE_4(k-1) = SSE_4(k-2)$, the above step size adaptation cease, and $r_{k+1} = r_k$. Better splitting results are expected, because, $SSE_4(k)$ is in a quadratic form, there exist only one local minimum in each splitting. Once it falls in the neighborhood region about the local minimum, the change in $SSE_4(k)$ becomes relatively small. Hence, it leads a small change in the orientation of the hyperplane's normal vector according to eq.(4)

3 Tree Search VQ

After clustering the input data into the desired number of clusters, the centroids of each partitioned cluster forms the VQ codebook. Noted that a Bentley K-d search tree is generated in the clustering process which can facilities efficient VQ. Further noticed that the partition planes are perpendicular bisectors of neighboring quantization vectors. As a result, hyperplane related misclassification is eliminated, as compared to the K-d tree based VQ in [8].

4 Image Coding Results

The performance of the proposed VQ design method is evaluated by VQ image coding. $256 \times 256 \times$ 8-bit grey level images are quantized by VQ of vector size 4×4 with codebook size 256. Such that the image compression ratio is 8 with the application of mean residue VQ. Three different training sets are used in the simulation to facilitate inside and outside training image coding performance analysis. The training set *Set 1* consists of 28672 training vectors extracted from 7 portraits. *Set 2* contains vectors from building images. *Set 3* contains all vectors from *Set 1* and *Set 2*. The objective performance of the generated codebook is measured by the peak signal to noise ratio (PSNR) of the image coding results.

$$PSNR = 10log_{10} \frac{255^2}{\frac{1}{256^2}\sum_{i=0}^{255}\sum_{j=0}^{255}(x_{i,j} - \hat{x}_{i,j})^2}(dB)$$

where $x_{i,j}$ and $\hat{x}_{i,j}$ are $(i,j)^{th}$ pixel of the original and the decoded image respectively.

The total SSE of the VQ codebook generated from various discussed algorithms are listed in Table 1. All the simulations are performed with the first hyperplane perpendicular to the principal component of the covariance matrix of the cluster data and passing through either O, P or Q that associated with minimum SSE. Algorithm I in Table 1 split the input data into 256 hyperboxes without hyperplane location adaptation. While Algorithm II performs hyperplane location adaptation with an *Iteration 1* only. The proposed algorithm with a fixed step size $r = 10^{-4}$ performs both hyperplane location and orientation adaptation.

Observed from Table 1, the total reduction in SSE performance of Algorithm I heavily depends on the input training sequences. Table 1 also showed that the maximum reduction in SSE by hyperplane location adaptation will lead to a large reduction in the total sum of SSE after 256 codevectors are generated (the results of Algorithm II). Observed from the last row in Table 1, the largest reduction in the total sum of SSE can be achieved by the proposed algorithm. Table 2 showed that the codebook obtained by maximum reduction in SSE provides better image coding performance in PSNR for all image training sets under consideration. The image coding results obtained from codebook generated by Algorithm II have an average of 0.1 dB improvement in PSNR when compared with that of Algorithm I. The proposed algorithm leads to the largest reduction in the total sum of SSE. Better VQ codebook is generated as observed that there are an additional 0.07 dB improvement in average when compared to that of Algorithm II. Moreover, codebook trained from different training sequences using the proposed algorithm have also showed similar improvement.

The performance is also consistent with other codebooks which are trained from different training sequences. Better image coding results as shown in Table 2. It showed that the proposed alogorithm is to achieve 0.02 dB improvement than that of algorithm II. This reveales the proposed algorithm is computational more efficient than that of other algorithms.

Table 3 and 4 show that the performance of the codebook generation when the adaptive algoritm for r is used. Introducing iterative algorithm in eq. (4) in the codebook generation, we have an additional 0.04 dB improvement in average for the images inside both training *Set 1* and *Set 2*. In addition, the adaptive algorithm in r leads a further reduction in total SSE.

5 Conclusions

An arbitrary hyperplane codebook generation and search algorithm is proposed in this paper. An iterative algorithm that search for the best adaptive hyperplane location and orientation for maximum reduction in sum of square error cluster partition has been derived. The algorithm is initiated by a hyperplane perpendicular to the principle component vector of the input data covariance matrix and multiple hyperplane location candidates (the centroid, RSWM, and median). The generated hyperplane is a perpendicular bisector of the centroids of the partitioned cluster sets. Simulation results on vector quantization image coding are presented to demonstrate the performance of the proposed codebook generation algorithm, which showed that the PSNR of the image coding results using codebook generated from the proposed algorithm outperforms that obtained from other algorithms in literature.

References

[1] Y.Linda, A.Buzzo, and R.M.Gray, "An algorithm for vector qunatizer design", *IEEE Trans. COM*, Jan.1980, pp. 84-95.

[2] S.Z.Selim and M.A.Islam, "K-Means Type Algorithms: A Generalized Convergence Theorem and Characteriszation of Local Optimality", *IEEE Trans. Pattern Analysis Machine Intelligence*, 1984, vol 6, pp. 81 - 87.

[3] S.C.Chan, Ted C.W.Kok, S.W.Chau, "Codebook Generation and Search Algorithm for Vector Quantization Using Arbitrary Hyperplanes" *IEEE International Symposiumon Circuits and Systems*, 1993, vol 3, pp. 1885 - 1888.

[4] J.Wan et al, "An algorithm for multidimensional data clustering", *ACM Trans. on Mathematcial Software*, vol.14, June 1988. pp.153-162.

[5] C.K.Ma and C.K.Chan, "Maximum descent method for image vector quantization", *IEEE Electronic Letters*, vol 27, no. 19, Sept. 1991, pp. 1772-1773.

[6] C.Y. Yang, C.C. Thien, anc J.C. Lin "Adaptive RWM-Cut: A New Fast Two-Class Clustering Method" *Pattern Recognition and Image Analysis*, vol9 , No. 3, 1999, pp. 465 - 472

[7] Kam-Tim Woo and Chi-Wah Kok, "Codebook Generation using Adaptive Arbitrary Hyperplane Algorihtm," *submitted for ISIT-2002*, Oct, 2001.

[8] D.Green, F.Yao and T.Zhang, "A Linear Algorithm for Optiaml Context Clustering with Application to Bi-level Coding," *Image Processing Proceeding*, vol.1, 1998, pp.508-511.

Set	Inital SSE $\times 10^7$	Total Sum of SSE ($\times 10^7$) in Algorithm		
		I	II	proposed
1	9.0218	2.8550	2.8287	2.8246
2	11.6750	2.9207	2.8354	2.8340
3	20.6990	6.0030	5.9460	5.9455

Table 1: Total SSE obtained by various VQ algorithms for different training sequences.

Set	Still Image		PSNR (dB) of Algorithm		
			I	II	Proposed
1	Inside	Lena	31.13	31.09	30.09
		Tiffany	33.08	33.18	33.22
		Zelda	34.11	34.16	34.22
		Baboom	25.41	25.42	25.43
	Outside	Peppers	29.24	29.45	29.48
2	Inside	Boat	29.13	29.23	29.21
		Bridge	26.32	26.39	26.40
		Goldhill	30.40	30.50	30.50
		House	33.37	33.54	33.55
		Clock	31.21	31.45	31.47
	Outside	Peppers	29.82	29.81	29.81
3	Inside	Lena	30.84	30.97	30.97
		Tiffany	33.07	32.99	33.03
		Woman	35.18	35.22	35.27
		Zelda	34.03	34.17	34.18
		Baboom	25.16	25.18	25.19
		Airplane	29.03	29.03	29.05
		Clock	31.06	31.15	31.17
	Outside	Peppers	29.98	30.15	30.14

Table 2: PSNR of VQ encoded images using codebook generated from different algorithms.

Set	Inital SSE $\times 10^7$	Total Sum of SSE ($\times 10^7$) in Proposed Algorithm with	
		fixed r	adaptive r
1	9.0218	2.8246	2.8060
2	11.6750	2.8340	2.8330

Table 3: Total SSE obtained by proposed algorithm with fixed and adaptive step size, r.

Set	Still Image		PSNR (dB) of Algorithm with	
			fixed r	adaptive r
1	Inside	Lena	30.09	31.15
		Barb	28.59	28.72
		Baboom	25.43	25.49
	Outside	Peppers	29.48	28.31
2	Inside	Airplane	29.38	29.38
		Boat	29.21	29.25
		Bridge	26.40	26.41
		Clock	31.47	31.48
	Outside	Peppers	29.81	29.68

Table 4: PSNR of VQ encoded images using codebook generated from proposed algorithm with and without adaptive stepsize, r.