Proceedings of the 5th WSEAS Int. Conf. on SIGNAL, SPEECH and IMAGE PROCESSING, Corfu, Greece, August 17-19, 2005 (pp5-10)

# An Algorithm for Vector Quantization Using Density Estimation

A. NONGNUCH and A. SURARERKS
Engineering Laboratory in Theoretical Enumerable System (ELITE)
Department of Computer Engineering
Chulalongkorn University
254 Phyathai Road, Patumwan, Bangkok
THAILAND

*Abstract:* - We are interested in the vector quantization problem. Many researches focus on finding a codebook using spatial distance. In order to evaluate the codebook, distortion is usually used. In this work, we introduce a novel concept for generating a codebook. Density estimation plays an important role to create codewords combining with a cluster distance. A vector quantization algorithm is proposed in this paper, some experimental results show that the output codebook can express the density distribution of the original vectors. Finally, three-dimensional vector problem is also considered.

*Key-Words:* - vector quantization, LBG algorithm, competitive splitting technique, distortion.

## 1 Introduction

Vector Quantization (VQ) is a lossy data compression method for approximating vectors which is widely used in various applications such as data compression and pattern recognition. The concept is close to that of "rounding-off". A small number of vectors, called *codewords*, are generated in order to be the representatives of all vectors. The set of all codewords is also called a *codebook*. Sometimes codebook can be considered as the encoded of the problem. Vector quantization can also be applied in the clustering problem. The result codewords are used to identify clusters.

The classical algorithm for vector quantization [1] has been proposed in 1980 by Linde, Buzo and Gray. This technique is named LBG algorithm which is originated by Least-squares quantization [7]. The key topic in vector quantization is a codebook design. It is usual for optimization method that the quality of a result codebook depends on the quality of initial one. In order to automatically perform a LBG algorithm, the extended version of LBG is studied in [4]. Two important techniques that are used to initialize codewords are Voronoi partition technique and centroid calculation method. A simple way of codebook initialization is a random method but it usually generates a poor initial codebook. In order to obtain a good one, several researches in literature have proposed techniques for codebook initialization such as splitting, maximum distance, competitive splitting [1,2,3] and self-organization map in [5,6].

For splitting technique, a codeword $C_i$ starts from the centroid or center of every vectors and splits it into two codewords $C_i + \varepsilon$ and $C_i - \varepsilon$. Splitting process continues till the number of codewords is reached. Codebook size in this method must be power of two. Different from splitting, maximum distance can give an arbitrary number of codewords. The first codeword starts from the maximum norm vector. A next codeword is a vector that is the most distant from the previous codeword. The process will be terminated when codebook size is fulfilled. Competitive splitting adopts splitting and competitive learning. For iteration, a vector will be randomly selected for competition. The codeword closed to this vector will be increased activation level. Codebook will be constructed from the activation level collecting competition between each codeword. The objective of those methods is to reduce distortion quantization that is a usual characteristic of quantization but there is still another well-informing characteristic.

Density (*i.e.*, the number of vectors in each cluster) is a characteristic that can evaluate the performance of vector quantization. Density can inform not only the difference of weight between clusters but also how much severely each codeword impacts vector quantization. A codebook can represent the density distribution of a problem; however, using a codebook for representing density distribution of vectors has not been conducted.

In this paper, we introduce a novel algorithm of vector quantization using density estimation. This algorithm is not only used for initializing LBG method but it can also be used as vector quantization

algorithm. Intuitively, codewords can represent the density distribution of vectors. Experimental results show that the density distribution of vectors can give information about the distribution structure of vectors as well. To evaluate density property, we define the variance of density (see detail in Section 3). In particular, the experimental results show that the new algorithm as initializer of LBG still converts in both distortion and density. After a brief of related works in Section 2, the new algorithm is proposed in Section 3. Some experimental results of the new algorithm comparing with random method and competitive splitting are studied in Section 4. Finally, the conclusion is discussed in Section 5.

# 2 Related Work

The summary of related works is described here.

## 2.1 LBG Algorithm

In 1980, LBG algorithm proposed in [1] or also known as GLA (Generalized Lloyd Algorithm) is a well-known vector quantization algorithm. Giving a good result, the main shortcoming of LBG is high computational complexity. Its inputs are the set of $P$ vectors and $\gamma$ threshold and its output is $N$-size codebook. The detail of this describes below:

1. *Codebook initialization*. $N$ vectors are generated as a codebook.
2. *Voronoi partition*. All vectors are assigned to the nearest codeword to make partitions.
3. *Termination condition check.* The change of previous iteration distortion and the current one will be considered. If $(D_{pres}\text{-}D_{curr})/D_{curr}$ is less than or equal to threshold $\gamma$, the algorithm will be terminated.
4. *Centriod calculation.* Each partition will be represented by its centroid
5. Go to step 2.

## 2.2 Competitive Splitting

In order to reduce quantization distortion, competitive splitting is a codebook initialization scheme introduced by [3] in 2004. By adopting splitting and competitive learning, this can allocate codebook according to spatial distribution of vectors. In each epoch, each codeword completes one another. The codeword that is the closest to a randomly selected vector will be the winner. Competitive learning is controlled by geometrical measurements as

$$ang(\mathbf{x}(t_c)-\mathbf{w}_c, \mathbf{x}(t_l)-\mathbf{w}_c) > \varphi_0 \text{ and}$$

$$\min(\|\mathbf{x}(t_c)-\mathbf{w}_c\|, \|\mathbf{x}(t_l)-\mathbf{w}_c\|) > d_0,$$

where $\mathbf{x}(t_c)$ is a selected vector for the current epoch, $\mathbf{x}(t_l)$ is a selected vector for the previous epoch and $\mathbf{w}_c$ is a codeword for the current epoch.

If a selected vector and the winner pass the thresholds of measurement, its activation level of the winner will be increased. If the activation level of the winner is greater than activation threshold $L_0$, the winner will be spitted. Although it adopts splitting technique, this can give an arbitrary-sized codebook by splitting only a codeword for each epoch. The detail of competitive splitting is described below:

1. Initialize the first codeword by the centroid of vectors, and set its activation level to 0.
2. Randomly take a selected $\mathbf{x}(t)$ from the set of vectors, find the winner $\mathbf{w}_c$ of the current competition in the set of codewords $\{\mathbf{w}_j\}(j = 1,2,\ldots,M(t))$, and decrease the activation level of all the losers as
$$L^{\mathbf{w}_j}(t) = (1-\lambda)L^{\mathbf{w}_j}(t-1), \text{ where } j \neq c$$
3. If the geometrical measurements of $\mathbf{w}_c$ satisfy the threshold, its activation level is increased by 1.0; otherwise, decrease the activation level as the losers.
4. If $L^{\mathbf{w}_c}(t) = 0 > L_o$, generate a new codeword $\mathbf{w}_{M(t)+1}$ from $\mathbf{w}_c$ by letting $\alpha_c$ be learning rate as
$$\mathbf{w}_{M(t)+1} = \mathbf{w}_c + \alpha_c(\mathbf{x}-\mathbf{w}_c),$$
and set $L^{\mathbf{w}_c}(t) = 0$ and $L^{\mathbf{w}_{M(t)+1}}(t) = 0$; otherwise, just update $\mathbf{w}_c$ by
$$\mathbf{w}_c + \alpha_c(\mathbf{x}-\mathbf{w}_c).$$
5. For each $\mathbf{w}_j$, if its winning frequency $\gamma_j$ from its birth to the current competition step is less than the prespecified pruning rate $\beta$, *i.e.*, $\gamma_j < \beta$, then delete $\mathbf{w}_j$ from the set of codewords.
6. If the number of the codewords reaches the prespecified codebook size, stop splitting; otherwise, go to step 2.

For practice, [3] suggests that $\varphi_0$ is set as 90° and $d_0$ is dynamically adjusted as $\Delta d_0 = -(\mu_0 - \mu)d_0$, where $\mu$ is the relative increment of the codeword number and $\mu_0$ is the expected relative increment of the codeword number in period of epoch.

# 3 Vector quantization using density estimation

In many researches, codebook is usually generated in order to represent the distribution of a problem. Since codewords are allocated according to only spatial distribution of vectors, codewords cannot represent further information of vectors. For example, Fig.1 (a) and (b) illustrate two different problems which have the same codebook. The codebook does not show the different size of the two subproblems.

In our work, we focus on how a codebook can express the density of original vectors. Using the same problem, Fig.2 shows that the codewords are allocated according to density of problem. The new VQ algorithm is developed so that it can allocate a codebook according to density distribution.

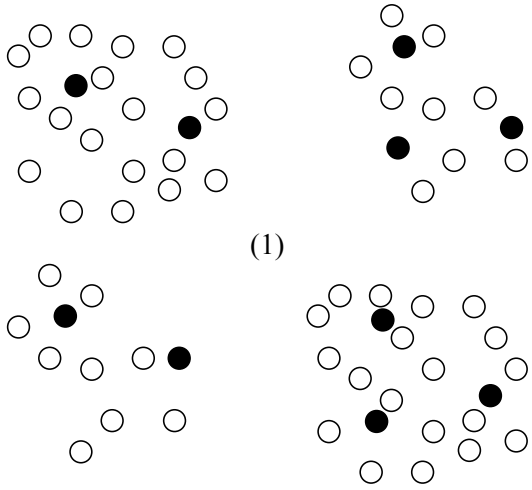



**Fig.1** Two different problems with the same codebook

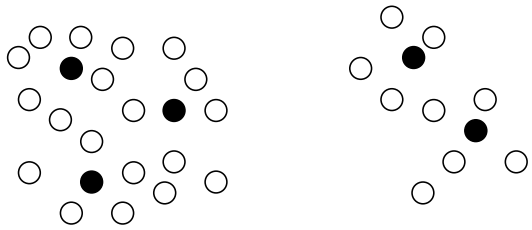


**Fig.2** Codebook representing the density

The concept of our algorithm is to partition nearly same- sized cluster, consequently density will be indeed kept. Roughly speaking this algorithm is splitting that try to keep size of each cluster equal to one another. The algorithm adopts divide and conquer technique. It continues recursively partitioning until the numbers of codeword reach the codebook size. Each codeword is the average value of vectors in each partition. Partitioning procedure will take a dimension of vectors. Dimensions chosen for partitioning is the dimension that can shapely distinguish between two clusters. Pivot is used for partitioning. In a dimension, a vector of which value in this dimension is greater than the pivot will be in a cluster and vice versa for the less. If vectors are partitioned into two clusters, left and right, the distance between the rightist value of vectors in left partition and the leftist value of vectors in right partition will be considered. This distance is called a *cluster distance*. The dimension of which cluster distance is the most will be chosen for each partitioning. Pivot is the average value of the rightist vector in the left partition and the leftist vector in the right partition. Fig.3 shows the example of partitioning where the gray gap indicates the cluster distance. The procedure of the proposed algorithm is described below:

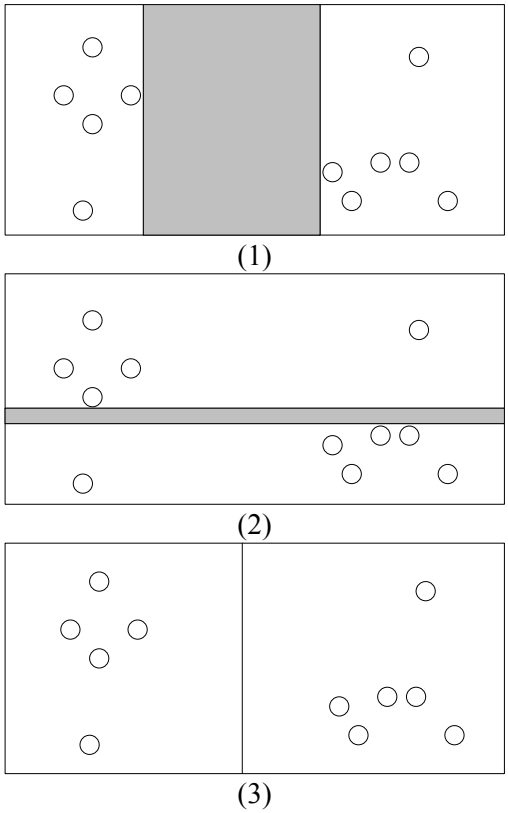



**Fig.3** Partitioning by cluster distance

**Algorithm:**

**Input:** $S$ the set of vectors
$N$ the codebook size

**Output:** $C$ the output codebook
*i.e.*, $C = \{ c_1, c_2, c_3, \ldots, c_N \}$

1. Let $S$ be a cluster.
2. Find *cluster distances* for every dimension of vectors in each cluster.
3. Partition each cluster into two nearly same-sized clusters according to the mid value of

vectors in the dimension with the most cluster distance and its pivot.

4. Continue partitioning till $N$ reached.
5. Generate a centroid for each partition and present them as codebook $C$.

To evaluate density property, we define *variance of density* as

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^{N}(n_{avg} - n_i)^2}{N}},$$

where $n_{avg}$ is the ratio of number of vectors and the codebook size, $n_i$ is the number of vectors per codeword $c_i$.

## 4 Experimental Results

This section shows some experimental results of the new algorithm as LBG initializer and VQ algorithm. We also compare the performance with LBG using competitive splitting and random initialization. Distortion and variance of density are used for performance evaluation. For distortion in this research, it is the *mean square error* (MSE). Four testing sets of vectors, illustrated by Fig.4, are generated for experiments.

Set I Two-dimension continuous uniform distribution vectors in range [0,1].

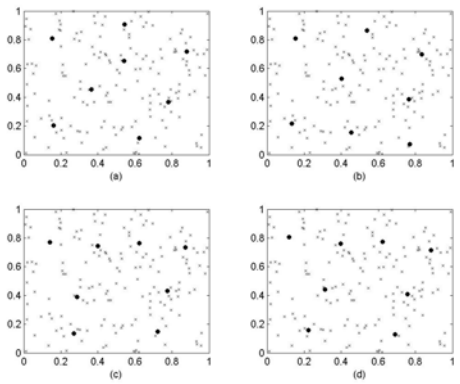
Set II Pairs $(x',y')$ sampled from sine curve $y = \sin(x)$, where $x \in [0,4\pi]$.

Set III Four different sizes Gaussian distribution clusters of two-dimension vectors with one variance and (1,1), (1,10), (10,1) and (10,10) mean for 10, 50, 70 and 30 vectors respectively.

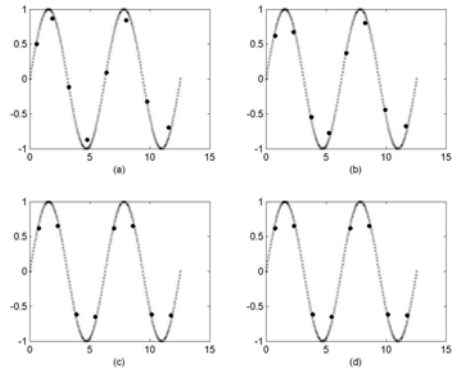
Set IV Four same sizes Gaussian distribution clusters of two-dimension vectors with the same mean and variance of set III.

Size of testing set I, II, III and IV are 160, 257, 160 and 160 vectors respectively. Distortion and variance of density are used for performance evaluation. Codebook sizes are four and eight. The threshold of LBG is 0.0005. Competitive splitting parameters are $\alpha_c = 0.2$, $\beta = 1/6$ and $\mu_0 = 0.05$. The results are shown in Table 1 and 2.

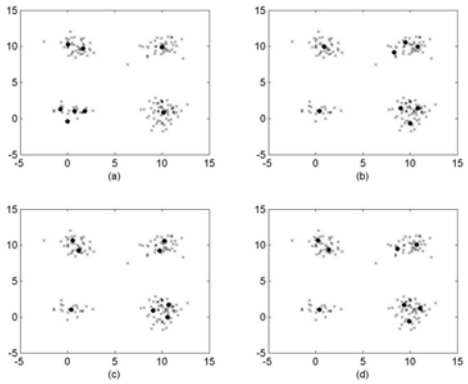
Table 1 shows the distortion of codebook obtained form our algorithm and LBG gotten initial codebooks form random method, competitive splitting and our algorithm on testing set I, II, III and IV respectively. For table 2, it shows the variance of density of codebook gotten form algorithms and testing set in the same Table 1.



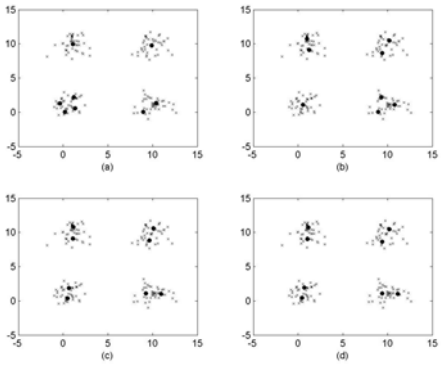
(1) Set I



(2) Set II



(3) Set III



(4) Set IV

**Fig.4** Four testing data sets with eight codewords

In Figs 4, the codebooks of each VQ algorithm are illustrated for four testing sets, where crosses are vectors and black points represent codewords. Finally, a codebook generation for 3D problem is also concerned in this work. The testing vectors are 3D helix of 400 vectors. The 32-size codebook is shown in Fig 5 for different views. The result shows that the new algorithm covers the problem.

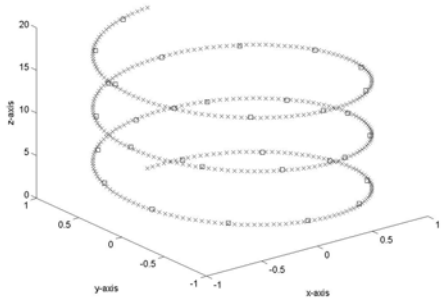
In order to test that codewords and vectors have the same distribution, chi-square goodness of fit test is applied. The results show that codewords obtained form the proposed algorithm and vectors in all testing set have the same distribution with significant 0.001.

**Table 1** Distortion comparison for four algorithms

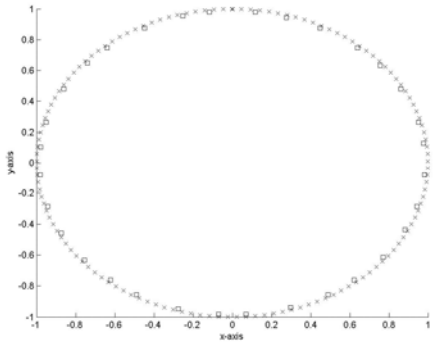
| VQ algorithm | Set I | | Set II | | Set III | | Set IV | |
|---|---|---|---|---|---|---|---|---|
| | Codebook Size | | Codebook Size | | Codebook Size | | Codebook Size | |
| | 4 | 8 | 4 | 8 | 4 | 8 | 4 | 8 |
| (a) Random LBG | 0.0404 | 0.0203 | 0.9249 | 0.3072 | 11.5735 | 1.6788 | 11.2291 | 1.3909 |
| (b) LBG with competitive splitting | 0.0404 | 0.0223 | 0.9249 | 0.3036 | 1.7430 | 1.1782 | 1.8368 | 1.1373 |
| (c) New algorithm | 0.0407 | 0.0225 | 0.9249 | 0.3030 | 4.9854 | 1.2612 | 1.8368 | 1.1222 |
| (d) LBG with new algorithm | 0.0404 | 0.0214 | 0.9249 | 0.3030 | 1.7430 | 1.0261 | 1.8368 | 1.1077 |

**Table 2** Variance of density comparison for four algorithms

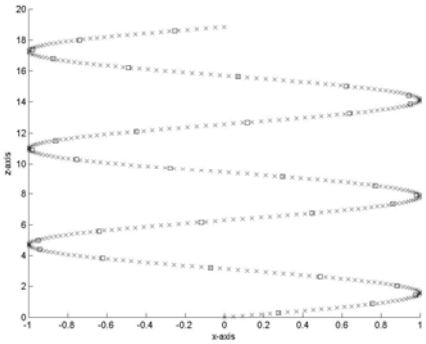
| VQ algorithm | Set I | | Set II | | Set III | | Set IV | |
|---|---|---|---|---|---|---|---|---|
| | Codebook Size | | Codebook Size | | Codebook Size | | Codebook Size | |
| | 4 | 8 | 4 | 8 | 4 | 8 | 4 | 8 |
| (a) Random LBG | 5.3852 | 7.0887 | 0.4330 | 5.6885 | 38.9230 | 19.1442 | 25.1396 | 13.3629 |
| (b) LBG with competitive splitting | 4.3012 | 5.5902 | 0.4330 | 3.1400 | 22.3607 | 9.4868 | 0.0000 | 9.3941 |
| (c) New algorithm | 1.8708 | 1.5811 | 0.4330 | 0.3307 | 17.7341 | 3.6742 | 0.0000 | 1.8028 |
| (d) LBG with new algorithm | 4.3012 | 4.3012 | 0.4330 | 0.3307 | 22.3607 | 3.3166 | 0.0000 | 3.0414 |



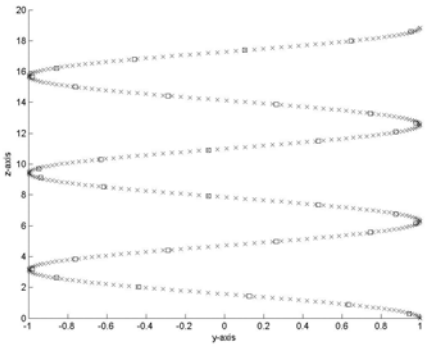

(1) Helix3D view




(2) x-y plane




(3) x-z plane




(4) y-z plane

**Fig.5** Codebook generation for 3D problem

# 5 Conclusion

This paper has proposed a novel algorithm for vector quantization. The idea of this algorithm is to maintain the density of each cluster a codebook represents; as the result a codebook can explain the density distribution of the problem. The experimental results show that the introduced algorithm can give an effectiveness codebook for variance of distribution on different problems.

*References:*

[1] Y. Linde, A. Buzo and R. M. Gray, An Algorithm for Vector Quantizer Design, *IEEE Transactions on Communications*, Vol.28, No.1, 1980, pp. 84-95.

[2] I. Katsavounidis, C.-C. J. Kuo and Zhen Zhang, A New Initialization Technique for Generalized Lloyd Iteration, *IEEE Signal Processing Letters*, Vol.1, No.10, 1994, pp. 144-146.

[3] Huilin Xiong, M. N. S. Swamy and M. O. Ahmad, Competitive Splitting for Codebook Initialization, *IEEE Signal Processing Letters*, Vol.11, No.5, 2004, pp. 474-477.

[4] G. Patane and M. Russo, Fully Automatic Clusering System, *IEEE Transactions on Neural Networks*, Vol.13, No.6, 2002, pp. 1285-1298.

[5] Lixin Xu, W. Q. Liu and V. Svetha, A Two-stage Vector Quantization Approach via Self-Organizing Map, *Proceedings of International Conference on Signal Processing*, Vol.1, 2002, pp. 913-916.

[6] T. Kohonen, *Self-organization and Associative Memory*, Springer New York, 1988.

[7] S. P. Lloyd, *Least-squares Quantizaton in PCM*, *IEEE Transactions on Information theory*, Vol.2, No.6, 2002, pp. 129-137.