

Enhanced Initialization Method for LBG Codebook Design Algorithm in Vector Quantization of Images

Kwok-Tung LO and Shing-Mo CHENG

*Department of Electronic Engineering
The Hong Kong Polytechnic University
Hung Hum, Kowloon, Hong Kong*

ABSTRACT

In this paper, a new initialization method is developed for enhancing the LBG codebook design algorithm in image vector quantization. The proposed method first arranges the training set data according to three different characteristics of the training vector, i.e. mean, variance and shape. A sampling method based on the criterion of maximum error reduction is then developed to select the desired number of representative vectors in the sorted training set as the initial codebook for the LBG algorithm. Computer simulations using real images show that the proposed approach outperforms the random guess and the splitting method. With the new approach, a higher quality of boundary preservation and a better local minimum are obtainable through a fewer number of iteration.

I INTRODUCTION

Vector Quantization (VQ) [1] has been widely known for its excellent rate-distortion performance. It may be simply viewed as a form of pattern recognition or matching where an input pattern is "approximated" by one standard template of a predetermined codebook. Conceivably, a good vector quantizer is subject to a good codebook. In general, if there is a codebook that satisfies the nearest neighbor, the centroid and the zero-probability boundary conditions, it is widely believed that the quantizer is at least locally optimal. However, since there is a variety of ways to partition a training set to define a vector quantizer, it is of great value to introduce a controllable mechanism to guarantee a better partition that leads to a better local optimum.

The most commonly used method for designing a VQ codebook is the LBG algorithm [1]. It is an iterative method and involves creating a primitive seed of letter reference from which an improving alphabet evolves step by step. The performance of the overall codebook depends on the selection of the initial seed (codebook). This initial seed could be a code generated arbitrarily or adopted previously.

Nevertheless, the almost exclusively adopted method is called the splitting algorithm[1], which grows increasingly larger codebooks of a fixed dimension from the lowest resolution codebook of a training set to the codebook of size as required for the initial seed. And, the simplest but less reliable initialization is by means of the so-called random guess method [1]. While the former method guarantees a high quality codebook and the latter suggests a fast approach for initial code selection, we propose in this paper a better and faster initialization method for LBG codebook design.

In the following, Section II presents the proposed LBG initialization method. Next, the complexity of the related algorithm is evaluated in section III and the simulation results are discussed in section IV. Lastly, some concluding remarks are given in section V.

II THE PROPOSED LBG INITIALIZATION

One of the main feature of LBG algorithm is the fulfillment of the nearest neighbor clustering. Basically, for a given set C of N output levels Y_n , $n = 1, 2, \dots, N$, the optimal partition cells R_p , $p = 1, 2, \dots, N$, satisfies $R_p \subset \{ X : d(X, Y_p) \leq d(X, Y_q) \}$; for all $q \neq p$, where $d(A, B)$ means the distortion between A and B. Accordingly, as the LBG process proceeds, it iteratively optimizes the minimum distortion mapping such that

$$d(X, Q(X)) = \min_{Y_p \in C} d(X, Y_p).$$

Conventionally, the mean squared error criterion is used as a distortion measure to define the best matching, that is

$$\begin{aligned} \min_{Y_p \in C} d(X, Y_p) &= \min_{Y_p \in C} \|X - Y_p\|^2 \\ &= \min_{Y_p \in C} \sum_{i=1}^k (x_p - y_{p,i})^2 \end{aligned}$$

Therefore, the whole LBG process intrinsically bears the target of approaching a zero Euclidean distance condition which results in the following derived relations :

$$\begin{aligned}
& \sum (x_i - y_i)^2 \rightarrow 0 \\
& \Rightarrow x_i - y_i \rightarrow 0 \\
& \Rightarrow x_i \rightarrow y_i \\
& \Rightarrow \sum x_i / k \rightarrow \sum y_i / k \\
& \Rightarrow \mu_X \rightarrow \mu_Y \quad \text{————— (1)}
\end{aligned}$$

where μ_A = the mean of A;

$$\begin{aligned}
& \Rightarrow x_i - \mu_X \rightarrow y_i - \mu_Y \\
& \Rightarrow \sum (x_i - \mu_X)^2 / k \rightarrow \sum (y_i - \mu_Y)^2 / k \\
& \Rightarrow \sigma_X^2 \rightarrow \sigma_Y^2 \quad \text{————— (2)}
\end{aligned}$$

where σ_A^2 = the variance of A;

In addition, as $x_i \rightarrow y_i$ and $\mu_X \rightarrow \mu_Y$, we have

$$\begin{aligned}
& \sum (x_i - \mu_X)^2 \geq 0 \\
& \Rightarrow \sum (x_i - \mu_X) (x_i - \mu_X) \geq 0 \\
& \Rightarrow \sum (x_i - \mu_X) (y_i - \mu_Y) \geq 0 \quad \text{————— (3)}
\end{aligned}$$

Thus, for a given partition, the relations (1), (2), and (3) are to be the necessary sub-conditions for the optimality. They may be respectively explained as the similarities of mean, variance and shape among the members of the same cluster. Based on these results, the training set may be pre-organized in favour of the LBG clustering process.

A. Hierarchical Arrangement of Training Set Data

In our work, each training vector is first allocated a hierarchical number composed of three parts: mean class, shape type and variance. The mean class is dependent on the mean value of the vector. We divide the vector into one of the eight different gray-level groups each of which constitutes a disjoint sub-set of the 256 gray levels. Meanwhile, the vector is also classified into one of eight shape type based on the bit-map of the vector. The bit-map of the vector is formed by comparing each component of the vector with its mean value. If it is greater than or equal to its mean, bit '0' is assigned; otherwise, bit '1' is set. Then, the resultant bit map of the vector is checked against the eight prescribed shape templates in Figure 1 by means of 'XOR' operations. Accordingly, the vector is assigned to the shape type that leads to the least number of output '1' by the 'XOR' operations. Finally, we simply assign the variance of the vector as the final part of the hierarchical number.

1 1 1 0	1 1 1 1	0 1 1 1
1 1 0 0	1 1 1 1	0 0 1 1
1 0 0 0	0 0 0 0	0 0 0 1
0 0 0 0	0 0 0 0	0 0 0 0
Type 1	Type 2	Type 3
1 1 0 0		0 0 1 1
1 1 0 0		0 0 1 1
1 1 0 0		0 0 1 1
1 1 0 0		0 0 1 1
Type 8		Type 4
0 0 0 0	0 0 0 0	0 0 0 0
1 0 0 0	0 0 0 0	0 0 0 1
1 1 0 0	1 1 1 1	0 0 1 1
1 1 1 0	1 1 1 1	0 1 1 1
Type 7	Type 6	Type 5

Figure 1. The eight template types

After number assignment, all the training vectors are sorted by their hierarchical numbers. With this arrangement, not only similar blocks are drawn nearby, but also blocks of unmatchable gray-levels are kept somehow separated. To a certain degree, this arrangement has already been an untrained prototype of the LBG clustering.

B. Initial Codebook Selection

After the hierarchical sorting, the whole training set is evenly divided into 4 sub-ranges in each of which the exact median vector is abstracted as the transient codeword of that sub-range. Then, the sub-range having the maximum mean squared distortion within the whole training set is to be further divided into 4 even sub-ranges of smaller size. And similar sub-divisions repeat until the required number of codewords is reached. With this maximum distortion reduction criterion, sub-ranges of high variance are likely to be further sub-divided. As a result, more representatives will be readily assigned to the partition cells which bear much texture information.

In addition, the LBG algorithm is also exploited at least in two aspects. Firstly, it is believed that the number of iterations may be decreased as the initial partition is to be much similar to its resultant one. Secondly, the chance of combining widely differential blocks into the same partition cell will be declined such that the average distortion may be further pressed down.

The details of the proposed initialization algorithm is shown as follows:

- Step1:* Sort the training vectors by their hierarchical numbers;
- Step2:* Divide the set of data into 4 sub-ranges, each sub-range has the same number of training vectors;
- Step3:* Choose the median vector of each sub-range to be the codeword for the sub-range;
- Step4:* If the number of codewords is equal to the required value, stop the process;
- Step5:* Select the sub-set having the maximum mean squared distortion to be further divided into 4 new sub-ranges, goto step 2.

In the process, there are two points worth mentioning. First, the number 2^s of sub-cells to be generated each time is so selected that $(n-2^s)/(2^s-1)$ is a positive integer as required for satisfying the condition of a given codebook size n . Second, the median vector of each sub-range is chosen as the substitute for the centroid as the codeword of the sub-range. This is not only because a fast assessment of codeword is advisable but also due to the fact that the replacement of the centroid by an approximated vector taken directly from the training sub-range may improve the overall VQ performance as proved in [2]. In principle, the best approximation is the closest vector that minimises the *MSE* between the vector and the centroid; but the search for this kind of approximation is nothing easier than the calculation of the centroid. Nevertheless, the centroid of a sorted set tends to be quite similar to the median member of the set. Therefore, we may let the median vector be a good approximation of the required codeword of each sub-range.

III COMPLEXITY OF THE INITIALIZATION

As to the valuation of the complexity of the above process, the worst case is considered. At first, we begin with 2^s partition cells of $m/2^s$ training vectors each, where m is the size of the training set. Then, we further divide the partition cell of maximum distortion into 2^s new sub-cells of $m/2^{2s}$ training vectors each. In principle, if the size of the partition cell to be chosen for sub-division is always to be the maximum of a certain stage, we will then have the worst case for maximum distortion computations. Therefore, suppose we have 2^s cells of $m/2^s$ members each in the first round, then we have $2^s \times m/2^s = m$ distortion computations at the beginning. And in the second round, we have in turn each cell obtained in the first round to be subdivided as it will occur in the worst case of maximum distortion computations. Consequently, we have $2^s \times$

$2^s = 2^{2s}$ cells of $m/2^s/2^s = m/2^{2s}$ members each and $2^{2s} \times m/2^{2s} = m$ distortion computations in the second round and so on. In general, given the required codebook size n , we have $\log_2 n / \log_2 2^s$ partition rounds for the initialization. Compared with the total number of $m \times n$ distortion computations in one LBG iteration, we have at most or at worst only $m \times [(\log_2 n / \log_2 2^s) - 1]$ distortion computations throughout the whole proposed initialization process.

IV SIMULATION RESULTS

We compare the proposed method with the two common LBG initialization algorithms, namely splitting and random guess, in terms of number of iterations and PSNR performance of the resultant codebook. Test images Building, Pepper and Tank are of size 256 x 256 and the codebook size is set to be of 256 16-dimensional codewords.

The performance of the LBG process with a threshold distortion rate fixed at 0.001 is demonstrated in Figure 2. Apparently, it takes fewer iterations for the proposed method to converge the improving codebook to its final stage. Although all the starting PSNR levels obtained by the splitting technique are higher than those by the proposed one, the subsequent PSNR levels of the latter go above those of the former. From another point of view, the distortion convergence rate of the proposed technique is faster. On the other hand, the performance of the random guess is the worst of all, though it has been widely accepted as the simplest one.

Test Images (Initialization)	No. of Iteration	MSE	Time (sec)
BUILDING			
Random Guess:	14	77.6	856
Splitting:	7	71.2	492
Proposed method:	5	68.6	371
PEPPER			
Random Guess:	17	95.7	1028
Splitting:	16	87.7	970
Proposed method:	12	83.5	729
TANK			
Random Guess:	14	306.8	847
Splitting:	12	288.1	720
Proposed method:	9	280.2	550

Table I Summary of LBG codebook design comparisons achieved by random guess, splitting, and the proposed method.

The summarized results of the test images are presented in Table I showing codebook design

comparisons of the LBG algorithm with random guess, splitting and the proposed method respectively. Obviously from the table, even though the number of iterations of the proposed method is the least among the three kinds of initialization, the performance in terms of MSE is the best in each case. And the computational time is also improved because of less iterations involved in the new method.

V CONCLUSIONS

In this paper, a simple LBG initialization is proposed. Instead of allowing a free developing initialization, we sample the training set based on its sorted statistics and the maximum distortion reduction criterion. With the introduction of a better initial partition which is as close as that of its final stage of the LBG design process, the new approach presents a faster and better local minimum than the splitting and random guess method.

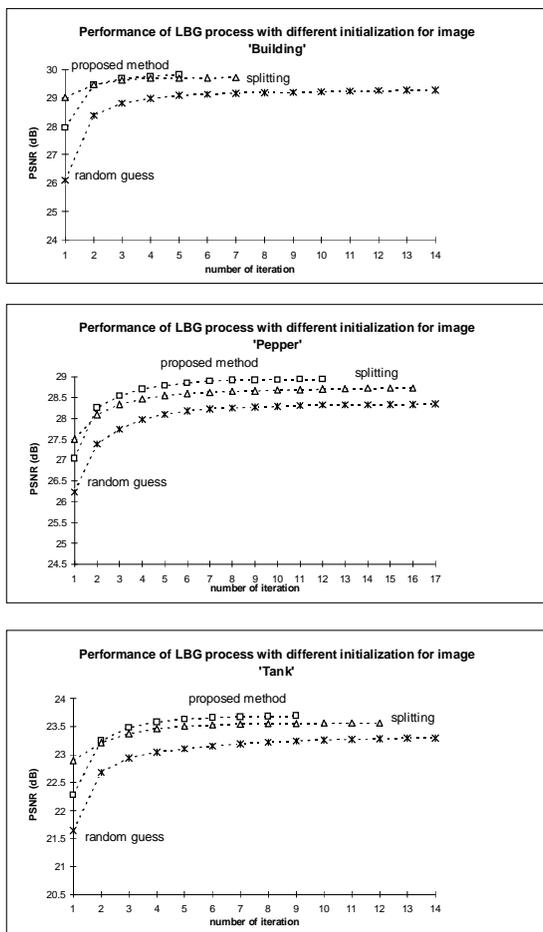


Figure 2. Performance of the LBG process with different initialization for test images

Figure 3a (top) is a highlighted portion of the original image Tank; Figure 3b is that of the same image reproduced by the random guess; Figure 3c is that by the splitting method and Figure 3d (bottom) is that reproduced by the proposed technique. It is apparent that the outlines of the star and the number 12 in Figure 3b and Figure 3c are rather vague. While in Figure 3d, the edge and texture detail is better preserved. As far as the ability to maintain a high fidelity to the source is concerned, it is evident that the proposed technique outperforms the conventional methods.

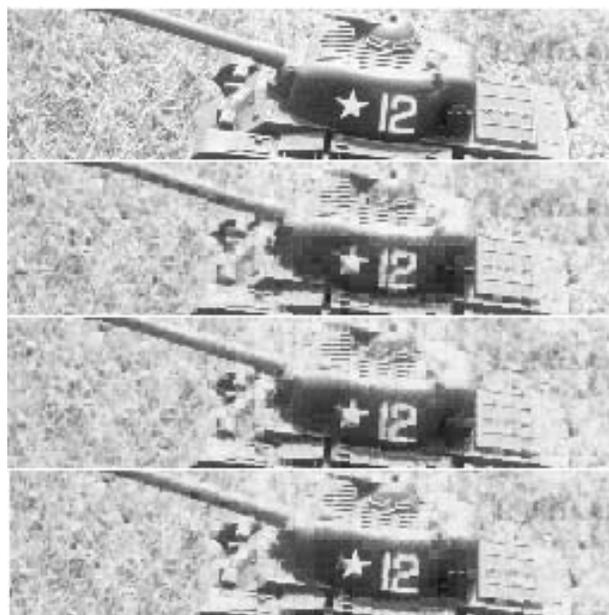


Figure 3 Highlighted portion of Image Tank of 3a (top) the original, 3b the "Random Guess" method, 3c the "Splitting" technique, and 3d (bottom) the proposed technique.

REFERENCES

- [1] A.Gersho and R.M Gray, *Vector quantization and signal compression*, Kluwer Academic Publishers, 1992.
- [2] H.A. Monawer, 'Image vector quantization using a modified LBG algorithm with approximated centroids', *Electronics Letters*, vol.31, no.3, pp.174-175, Feb. 1995.