A NOVEL CODEBOOK INITIALISATION TECHNIQUE FOR GENERALIZED LLOYD ALGORITHM USING CLUSTER DENSITY

K.Somasundaram
Department of Computer Science and Applications
Gandhigram Rural Institute
Gandhigram – 624 302
Tamilnadu, India

S.Vimala
Department of Computer Science
Mother Teresa Women's University
Kodaikanal – 624 102
Tamilnadu, India

Abstract - **In this paper, a novel codebook initialization technique has been adopted for Generalized Lloyd Algorithm (GLA). GLA plays an important role in the design of Vector Quantizers. The proposed technique generates initial seeds in a simple way which reduced the computational complexity when compared to other existing techniques. The initial codebook thus generated using the proposed technique is iteratively improved using GLA to improve the quality (PSNR) of the reconstructed images**.

Keywords: codevector, training set; PSNR; codebook.

I. INTRODUCTION

Vector Quantization (VQ) is an efficient image coding technique achieving low bit rates i.e. lower than one bit per pixel (bpp). Several VQ algorithms have recently been extensively investigated/developed for speech and image compression. VQ is a lossy image compression technique and has applications in different areas: protein classification and secondary structure computation [1], speech recognition, face detection, pattern recognition, real time video based event detection and anomaly intrusion detection systems [2] etc.

The principle component of VQ is the codebook that is used for encoding the images. The whole image is split into small blocks of size 4 x 4. Each block is converted into a vector of size K. This vector is called a training vector and the set of training vectors is called a training set. A training vector is approximated by a representative codevector in the codebook. Compression is achieved by transmitting or storing the codeword address (index) instead of the codeword itself.

VQ finds its role in various compression schemes such as Sub-band coding, Predictive coding and even in Lossless compression algorithms. The performance of VQ is highly dependent on the optimality of the codebook.

Two basic tasks of VQ image compression are:
1. Design of the codebook and
2. The search for the best approximation for each block.

A vector quantizer Q of dimension k and size N is a mapping from a vector in k-dimensional Euclidean space points. This can be expressed mathematically as $Q: R^k \rightarrow C$, where $C=\{y_1, y_2, y_3, \ldots, y_N\}$, and $y_i \in R^k$. The set C is called the codebook and $y_i, \leq i \leq N$, are called the codevectors or codewords. To measure the performance of a vector quantizer, a distortion measure $d(x,Q(x))$ has to be defined in association with any input vector x and its reproduction vector $Q(x)$. With such a measure, one can quantify the performance of a vector quantizer by either the average distortion $D=E[d(x,Q(x))]$ or the worst case distortion $D_{max}=max_x d(x,Q(x))$. The design of the optimal quantizer is to seek the codebook that minimizes the average distortion over all possible codebooks.

There are several established methods for generating a codebook [3]. The most cited and widely used is the Generalised Lloyd Algorithm [4]. A different approach is to build the codebook hierarchically. A top-down process: Iterative Splitting Algorithm and a bottom-up approach: Pairwise Nearest Neighbor Algorithm [5]. But the computational complexity involved in these techniques is high when compared to that of GLA.

The most common GLA starts with initial codebook, where the codevectors are chosen arbitrarily. The generated initial codebook is then improved iteratively. Various methods are involved in creating the initial codebook for GLA: one such method is the Simple Codebook Generation (SCG) [6], where the desired number of codevectors is selected arbitrarily from the training set.

A. Simple codebook generation technique (SCG)

The codevector from every n^{th} position is selected from the training set to form the codebook of desired size M. The value for n is calculated as n = N/M, where N is the size of the training set and M is the size of the desired codebook to be generated.

B. Steps to generate the Codebook

Step1: Input the image
Step2: Split the images into small blocks of size 4 x 4.
Srep3: Generate the training set of size N.
Step4: Compute n=N/M.
Step5: Select the training vectors one by one from every n^{th} position till the desired codebook of size M is reached.

In this paper, a novel method using the Cluster Density is introduced. The rest of the work is organized as follows: In section 2, the proposed method is explained. In section 3, the results are discussed and the conclusion is given in Section 4.

II. PROPOSED METHOD

In this method, initially all the training vectors are treated as Cluster centers. The distance between a training vector X_i and all the other training vectors Y_j where $i \neq j$ is computed as

$$D = \sum_{j=1}^{k}\left(X_j - Y_{ij}\right)^2 \text{ where } X \neq Y_i \text{ and } 1 \le i \le N \qquad (1)$$

The minimum of all the distances is identified and the Cluster Density of the corresponding training vector is incremented by 1. For example, if the distance between the 1^{st} and the 250^{th} vectors is the minimum, then the cluster density of the 250^{th} vector is incremented by 1. Similarly the distance between the second training vector and all the other vectors are computed and so on. Whichever distance is minimum the cluster density of the corresponding vector is incremented by one. The above steps are repeated for all the training vectors one by one. Hence if 25 training vectors are closer to any training vector i, then the cluster density of i^{th} vector is 25.

The training vectors are sorted in the descending order based on their cluster densities. Finally the M training vectors with the top density values are selected as the seeds for the initial codebook that is to be used as the input for the GLA.

A. Steps to generate the codebook

Step1: Input the image
Step2: Split the image into small blocks of size 4 x 4.
Step3: Generate the training set of size N.
Step4: Initialize the Density array of size N with zero.
Step5: for i=1 to N
Step6: for j=1 to N
Step7: if i not equal to j
Step8: compute the distance between the vectors i and j using the equation (1).
Step9: next j
Step10. Find the minimum of all the N-1 distances computed and increment the corresponding density by 1.
Step11. next i.
Step12. Sort the training vectors in ascending order based on their density values.
Step13: Select the top M training vectors to form the initial codebook.

III. RESULTS AND DISCUSSION

We carried out the experiments using the SCG and proposed techniques on standard images Lena, Boats and Baboon, each of size 256 x 256 pixels and computed the PSNR values of the reconstructed images and the time taken for generating the codebook. The values are given in Table-1.

Table-1: Comparison of SCG and the proposed methods with respect to PSNR of reconstructed images and codebook generation time with codebooks of different sizes.

CB Size		Lena		Boats		Baboon	
		SCG	Proposed	SCG	Proposed	SCG	Proposed
128	PSNR	29.24	33.80	26.40	31.26	33.12	38.67
	Time	0.02	55.80	0.02	54.75	0.02	54.95
256	PSNR	30.66	34.82	28.33	32.77	34.36	39.74
	Time	0.02	54.97	0.02	54.77	0.02	55.70
512	PSNR	32.23	36.30	29.56	34.87	36.26	40.99
	Time	0.02	54.98	0.02	54.76	0.02	55.08
1024	PSNR	33.83	37.57	30.99	36.41	37.90	42.30
	Time	0.02	54.99	0.02	56.26	0.02	55.13
2048	PSNR	36.35	39.34	33.31	37.82	40.56	43.86
	Time	0.02	55.25	0.02	55.24	0.02	55.07

Codebooks of sizes 128, 256, 512, 1024 and 2048 are created using both the methods. It is inferred clearly from Table-1, that the PSNR values of the reconstructed images obtained using the proposed methods are improved significantly.

From Table-1, it is observed that the time taken by the proposed method is more when compared with that of the SCG method. But the quality of the reconstructed images obtained with the proposed method is significantly better than that of SCG. On an average, the PSNR of the SCG method is increased by 4.49 with the proposed method. The average PSNR values obtained with both the techniques are 32.87 and 37.37 respectively.

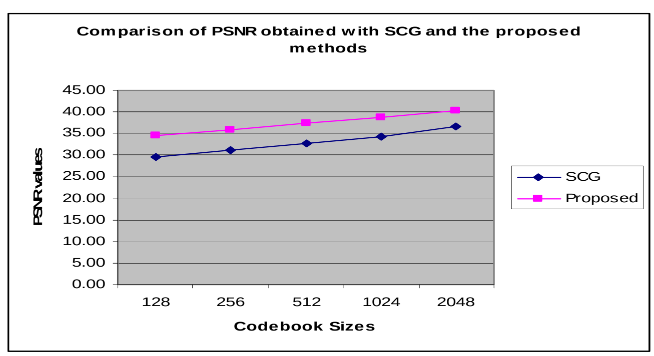

Fig.-1: Comparison of Average PSNR for different codebook sizes.

Figure 1 shows the comparison of the algorithms SCG and the proposed one with respect to PSNR by varying the codebook size.

It is further observed that our proposed method, on an average yields 12.04% increase in the PSNR value obtained by the existing SCG method. The proposed algorithm gives 16.87%, 14.98%, 14.39%, 13.20%, 9.80% and 13.67% increase in PSNR values for the codebooks of sizes 124, 256, 512, 1024 and 2048 respectively. Although we have compared our proposed algorithm with only one technique, it will be definitely efficient, simple and fast as compared to other codebook initialization algorithms such as Pairwise Nearest Neighbor (PNN), Split and Merge Techniques etc.

IV. CONCLUSION

A new codebook initialization algorithm is proposed. It is observed from the results that the proposed method gives better PSNR values. Generalized Lloyd Algorithm is the most widely used method for codebook generation for vector quantization. GLA needs initial codebook which is to be optimized by it. The proposed method is one such technique to generate the initial codebook for GLA by considering the density of the clusters. From the results it is observed that the quality of the reconstructed images obtained with the proposed method is good when compared to that of one other codebook initialization method SCG. The computational complexity involved is also less.

REFERENCES

[1] Juan J.Merelo Guervos, and et, al., *Protein Classification and Secondary Structure Computation,* Proceedings of the International Workshop on Artificial Neural Netwroks, Pages 415-421, 1991, Springer-Verlog.

[2] C.Garcia and G.Tzirita, *Face Detectionusing Quantized Skin Color Regions Merging and Wavelet Packet Analysis*, IEEE Trans. Multimedia, vol. 1, No. 3, pp. 264-277, September 1999.

[3] A.Gersho and R.M.Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

[4] Y.Linde, A.Buzo and R.M.Gray, *An Algorithm for Vector Quantizer Design*, IEEE Transactions on Communications, 28(1), 84-95, January 1980.

[5] P.Franti, T.Kaukoranta and O.Nevalainen, *On the Splitting Method for VQ Generation*, Optical Engineering, 36(11), 3043-3051, November 1997.

[6] K.Somasundaram and S.Vimala, *Simple and Fast Ordered Codebook Generation for Vector Quantization*, Proceedings of the National Conference on Image Processing, ISBN: 978-81-8424-574-5, Gandhigram Rural Institute – Deemed University, India, March 2010.