International Journal of Computer Applications (0975 – 8887)
Volume 78 – No.6, September 2013

K-Means Codebook Optimization using KFCG Clustering Technique

Tanuja K. Sarode. Ph.D
Associate Professor
TSEC
Mumbai, India

Nabanita Mandal
Lecturer
TSEC
Mumbai, India

ABSTRACT

Codebook Optimization is a concept of vector quantization which is applied to achieve lossy compression. Optimization of the codebook helps in maintaining the quality of the image. The codebook is generated using Kekre's Fast Codebook Generation (KFCG) algorithm and Random Selection Method. The K-means algorithm is used to optimize the codebook. The Mean Square Error (MSE) is used as the measurement parameter. The point where the MSE converges is the optimal point and the codebook at that point is said to be the optimized codebook. The results obtained show that when K-means algorithm is applied to the codebook generated by KFCG Algorithm, less MSE is obtained as compared to Random Selection Method.

General Terms

Image Processing, Compression, Optimization.

Keywords

Codebook Optimization, Euclidian Distance, K-means Algorithm, Mean Square Error, Vector Quantization

1. INTRODUCTION

In the era of internet, fast transmission of images has become a necessity. But fast image transmission requires high bandwidth. It is not always possible to get high bandwidth. So, image compression techniques are used which serves the purpose. In image compression, the irrelevant and redundant image data is reduced in order to store or transmit the image in an efficient way. Compression techniques can be classified as lossy and lossless [1]. Lossless compression techniques are suitable where each and every minute technical detail of the image is significant. For example, in the medical domain, if a very small part of the image is missing then it creates havoc. On the other hand, lossy compression methods are suitable for natural images where minor loss of fidelity is acceptable. One such lossy compression technique is Vector Quantization (VQ) [2]. In this technique, the image is represented in the form of vectors. VQ is a mapping function, Q that maps a vector in K-dimensional vector space, R^k into a finite subset of the vector space W containing N distinct vectors [3]. Hence, Q: $R^k \rightarrow W$.

In vector quantization, the image data is stored in a codebook. Each vector present in the codebook is called a codevector. A good codebook is very much essential for VQ. Once the codebook size is fixed, then after applying the different techniques for codebook generation, the Mean Square Error (MSE) is obtained. The distortion obtained from the different codebook generation algorithm varies even if the codebook size is same. But the minimum error is not achieved. The reason is that the codevectors in the codebook may not have reached their optimal position. When the codebook is optimized, the MSE reaches a value beyond which it cannot be reduced.

In literature, vector quantization has been successfully used for image compression. In Pamela C. Cosman et al's paper [4], they have given the fundamental idea of vector quantization and how it can be used for image compression. The process of VQ involves codebook design. S. Vimala et al. have used interpolation to generate the codebook [5]. R. Krishnamoorthi and N. Kannan has proposed orthogonal polynomial based transformation for codebook generation [6]. A survey of codebook generation techniques has been done by Tzu-Chuen Lu and Ching-Yun Chang. In their paper, LBG, Enhanced LBG, Neural Network based techniques, Genetic Algorithm based techniques etc. has been discussed [7]. After the codebook is generated, optimization of the codebook is done using various algorithms. H.B. Kekre and Tanuja K. Sarode has proposed K-means algorithm for optimization of codebook [8]. LBG [9] and KFCG [10] algorithm has been applied by them for generating the codebook and to optimize it, they have used K-means algorithm [11].

In this paper, the K-means Algorithm [11] which is a clustering technique, will be used on the codebook for optimization. In this technique, initially, the codevectors are selected randomly which doesn't lead to optimization. So, this process is repeated for many iterations and the MSE value is calculated for each iteration. The MSE reduces in each iteration. After some iterations, the MSE value converges. This point is the optimal point and the optimized codebook is obtained. The K-means Algorithm is applied to the codebook generated by KFCG Algorithm [10] and Random Selection method. The image is reconstructed back using the optimized codebook.

This paper consists of six sections. In Section 2 the proposed system has been discussed. Section 3 contains the description of the codebook generation algorithms. Section 4 consists of description of the optimization algorithm used. Section 5 provides the results obtained after application of various techniques. The conclusion is given in Section 6.

2. VQ IMAGE COMPRESSION SYSTEM

The VQ image compression system is shown in Fig. 1. The process of vector quantization consists of encoding and decoding phases.

The encoding and decoding phases can be described as follows:

A. Encoding Phase

The input image is converted into vectors by dividing the image matrix into 2×2 non overlapping blocks. The RGB components of each pixel are separated and values of each block are written together. This forms the codevector. The collection of all codevectors gives the training vector space of the image. From the training vector space, codebook is generated using different algorithms. The generated codebook is then optimized using the K-means algorithm. An index is created which keeps track of each codevector. The optimized codebook and the index are used to reconstruct the image.

B. Decoding Phase

In the decoding phase, the image is reconstructed back using the optimized codebook and the index. A new training vector space is created in the receiver's side which consists of the codebook values. The correct position is obtained using the index. The image is reconstructed back using this training vector space.

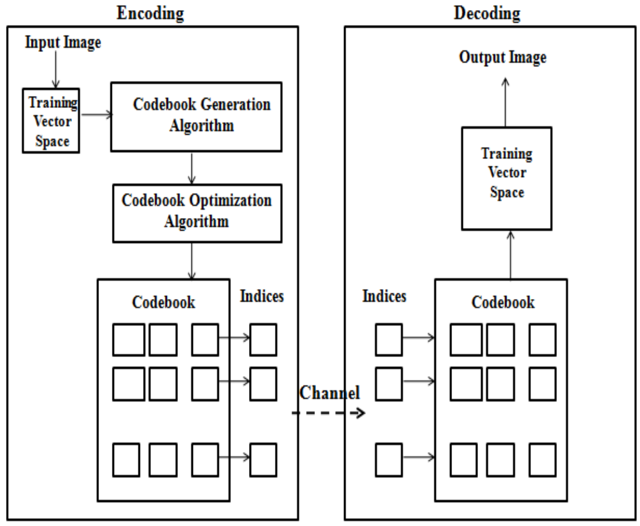


Fig. 1: VQ Image Compression System

3. CODEBOOK GENERATION ALGORITHMS

The technique applied for the generation of codebook is KFCG algorithm [10]. A Random Selection method has also been introduced for obtaining the codebook.

3.1 Kekre's Fast Codebook Generation Algorithm

Kekre's Fast Codebook Generation (KFCG) algorithm [10] uses comaprision between members of training vectors and codevectors. The codebook generation is faster. The training vector space is considered as a single cluster initially. The centroid obtained is the codevector. The 1st iteration of KFCG Clustering is shown in Fig. 2. C1 is the codevector obtained after centroid calculation. The first member of training vector is compared to the first member of codevector in the first iteration. The training vector is put in first cluster if its first member is less than the first member of codevector. Otherwise, the training vector is put in the second cluster. In the second iteration, the first cluster is split into two by comparing second member of the training vector with the second member of codevector. The 2nd iteration of KFCG Clustering is shown in Fig. 3. Again second cluster is split into two by comparing the second member of the training vector with that of the codevector. This procedure is repeated till desired codebook size is reached.

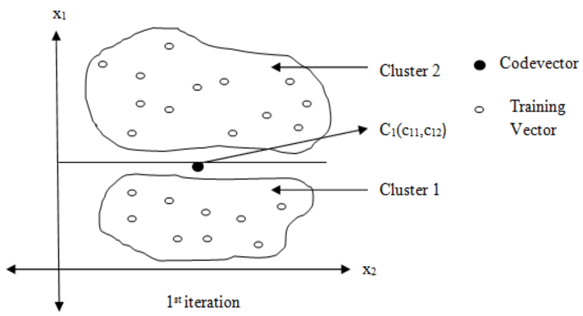


Fig. 2: 1st iteration of KFCG Clustering

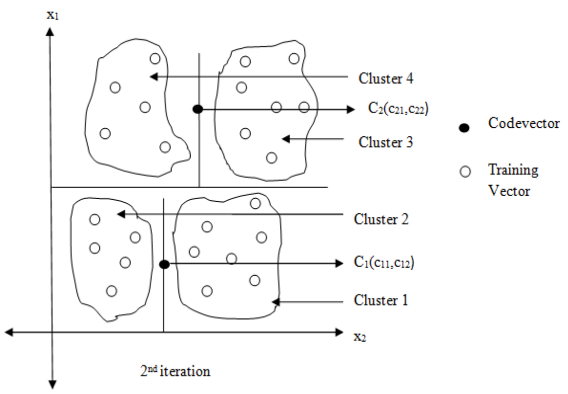


Fig. 3: 2nd iteration of KFCG Clustering

3.2 Random Selection Method

In this method, codebook of the desired size is obtained by random selection of vectors from the training vector space. In other words it can be said that the size of the codebook is dependent on the number of random vectors initially chosen.

The steps are as follows:-

1. Generate the training vector space, T of the image which contains M training vectors.
   T= $\{X_1, X_2, X_3 \ldots X_M\}$
   X_i is the training vector and it is represented as $X_i = \{x_{i1}, x_{i2}, x_{i3}, \ldots x_{ik}\}$
   where,
   k denotes the dimension.
2. Select K random vectors from the training vector space where K denotes the desired codebook size. These initial random vectors serve the purpose of the initial codebook.

The codebook obtained by KFCG Algorithm and Random Selection Method are fed as input to the K-means algorithm for optimization.

4. CODEBOOK OPTIMIZATION ALGORITHM

The K-means algorithm is used for optimizing the codebook obtained from KFCG algorithm and Random Selection Method.

4.1 K-means Algorithm

The K-means algorithm [11] is a clustering algorithm where K denotes the size of the cluster. The basic idea behind this algorithm is to form K clusters and assign each object to one of the K clusters in such a way that the measure of dispersion within the cluster is minimized [12-13]. The dispersion can be measured using the squared Euclidean distance.

The following formula is used to calculate the Euclidean distance:-

$$D\,(X_i\text{-}C_j) = \sum_{p=1}^{k} (x_{ip}-c_{jp})^2 \qquad (1)$$

where,
X_i is the training vector,
C_j is the codevector.

This algorithm aims at minimizing the objective function. Here, the mean square error is the objective function. So, the MSE reduces at each iteration and after the codebook is optimized, the MSE reaches a value after which it stops reducing. It converges at the optimal point.

In this algorithm, K-random vectors are selected from the training vector space which are called as codevectors. The squared Euclidean distance of all the training vectors with the selected K vectors are obtained and K clusters are formed. If the squared Euclidean distance of training vector Xj with i^{th} codevector is minimum then X_j is put in i^{th} cluster. Centroid of each cluster is obtained. The centroids of all clusters obtained in the previous iteration form the set of new codevectors which is the input to K-Means algorithm for the next iteration. The MSE for each of the K clusters is computed and net MSE is obtained. This process is repeated till the net MSE converges.

The MSE is calculated using the following formula:-

$$\text{MSE}\,(m) = \frac{1}{Zk}\sum_{r=1}^{Z} (X_r-C_m)^2 \qquad (2)$$

where,
m is the cluster number,
Z is the number of vectors in that cluster,
k is the dimension,
X_r is r^{th} training vector,
C_m is the codevector of m^{th} cluster

Instead of selecting K random vectors, the codebook of size K obtained by KFCG and Random Selection Method are used for optimization using the K-means algorithm.

5. RESULTS

The algorithms are implemented on Intel Core i5-3210M, 2.50 GHz, 4GB RAM machine and Matlab R2011b is used. We have applied these techniques on four color images each having size of 256×256×3.

Training Images of size 256×256×3 are shown in Fig. 4.

The mean square error and the number of iterations required for optimization of codebook obtained from KFCG algorithm and Random Selection Method on codebook size 512 and 1024 are shown in Table I and II respectively.

The sample initial and final images for codebook size 512 when KFCG algorithm is used are shown in Fig. 5.

The variation of MSE for codebook size 512 when KFCG algorithm is used is shown in Fig. 6.

(a) (b) (c) (d)

Fig. 4: Training Images (a) Tiger (b) House (c) Roses (d) Puppies

Table I. Result of K-means Algorithm on codebook size 512

S. no.	Image	Algorithm	Initial MSE	Final MSE	No. of Iterations
1.	Tiger	KFCG	206.11	67.43	90
		RANDOM	287.95	101.91	245
2.	House	KFCG	246.41	74.04	256
		RANDOM	282.61	118.17	191
3.	Roses	KFCG	533.34	127.15	148
		RANDOM	350.39	144.72	116
4.	Puppies	KFCG	179.49	56.03	141
		RANDOM	642.68	69.26	336

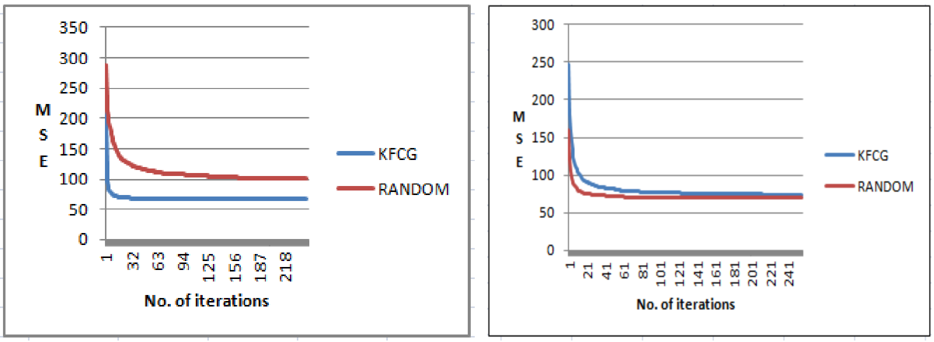
Table II. Result of K-means Algorithm on codebook size 1024

S. no.	Image	Algorithm	Initial MSE	Final MSE	No. of Iterations
1.	Tiger	KFCG	172.34	54.95	66
		RANDOM	232.43	94.57	176
2.	House	KFCG	231.46	81.22	227
		RANDOM	257.95	98.54	250
3.	Roses	KFCG	571.35	122.12	127
		RANDOM	519.31	153.84	173
4.	Puppies	KFCG	168.21	51.54	205
		RANDOM	607.13	62.62	244

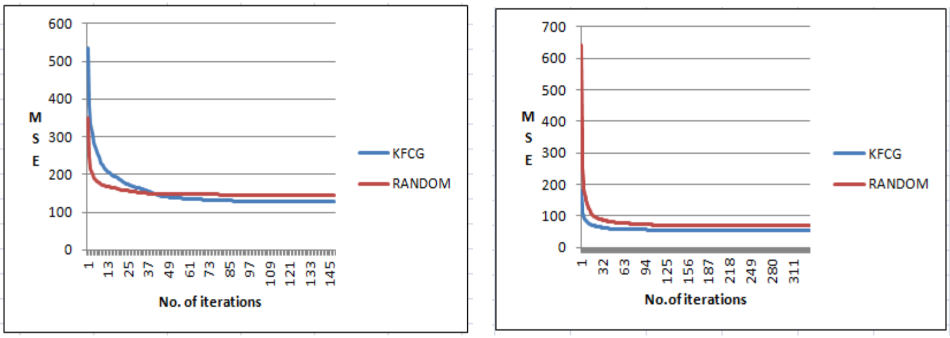
(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

Fig. 5: Initial and Final images with codebook size 512 (a) Initial MSE=206.11 (b) Final MSE=67.43 (c) Initial MSE=246.41 (d) Final MSE=74.04 (e) Initial MSE=533.34 (f) Final MSE=127.15 (g) Initial MSE=179.49 (h) Final MSE=56.03



(a) **(b)**



(c) **(d)**

Fig. 6: Variation of MSE (a) Tiger (b) House (c) Roses (d) Puppies

6. CONCLUSION

Codebook generation and optimization can be done using various algorithms. In this paper, codebook is generated using KFCG and Random Selection Method. The K-means algorithm is used on the generated codebook for optimization. We have varied the codebook size to show the change in mean square error. It is observed that, in most of the cases, the codebook obtained from KFCG algorithm gives less mean square error as compared to other algorithms. So, when the K-means algorithm is applied to the codebook obtained from KFCG algorithm, the best optimized codebook is obtained.

7. REFERENCES

[1] R. Navaneethakrishnan, "Study of Image Compression Techniques," International Journal of Scientific & Engineering Research, Vol. 3, No. 7, pp. 1-5, July 2012.

[2] G. Boopathy and S. Arockiasam, "Implementation of Vector Quantization for Image Compression - A Survey," Global Journal of Computer Science and Technology, Vol. 10, No. 3, pp. 22-28, April 2010.

[3] Carlos R.B. Azevedo, Esdras L. Bispo Junior, Tiago A. E. Ferreira, Francisco Madeiro, and Marcelo S. Alencar, "An Evolutionary Approach for Vector Quantization Codebook Optimization," Springer-Verlag Heidelberg, pp. 452-461, 2008.

[4] Pamela C. Cosman, Karen L. Oehler, Eve A. Riskin, and Robert M. Gray, "Using Vector Quantization for Image Processing," In Proc. Of The IEEE, Vol. 81, No. 9, pp. 1326-1341, September 1993.

[5] S. Vimala, K. KowsalyaDevi and M. Sathya, " Codebook Generation for Vector Quantization using Interpolations to Compress Gray Scale Images," International Journal of Computer Applications, Vol. 42, No. 9, pp. 14-19, March 2012.

[6] R. Krishnamoorthi and N. Kannan, "Codebook Generation for Vector Quantization on Orthogonal Polynomials based Transform Coding," International Journal of Information and Communication Engineering, Vol. 5, No. 1, pp. 67-73, 2009.

[7] Tzu-Chuen Lu and Ching-Yun Chang, "A Survey of VQ Codebook Generation," Journal of Information Hiding and Multimedia Signal Processing, Vol. 1, No. 3, pp. 190-203, July 2010. [8]

[8] H.B. Kekre and Tanuja K. Sarode, "Vector Quantized Codebook Optimization using K-Means," International Journal on Computer Science and Engineering, Vol. 1, No.3, pp. 283-290, 2009.

[9] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun.', Vol. COM-28, No. 1, pp. 84-95, January 1980.

[10] H. B. Kekre and Tanuja K. Sarode, "Fast Codebook Search Algorithm for Vector Quantization using Sorting Technique," International Conference on Advances in Computing, Communication and Control, pp. 317-325, 2009.

[11] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations", Proceedings of 5th Berkeley symposium on Mathematical Statistics and Probability", Berkely, University of California Press, vol.1, pp. 281-297, 1967.

[12] Yujun Lin,Ting Luo, Sheng Yao, Kaikai Mo, Tingting Xu and Caiming Zhong, "An Improved Clustering Method Based on K-means," In Proc. Of 9th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, pp. 734-737, 2012.

[13] Bhanu Sukhija and Sukhvir Singh, "Improved K-Means clustering technique using distance determination approach," International Journal of Advanced Research in Computer Science and Electronics Engineering, Vol. 1, No. 5, pp. 31-34, July 2012.