Analysis of Codebook Generation Techniques for Vector Quantization

Nutan Palshikar
II Year Student
Department of Computer Engineering,
MGM's College of Engineering and Technology,
Navi Mumbai
University of Mumbai, India.

P. S. Lokhande
Head of I.T. Department
Department of Information Technology,
MGM's College of Engineering and Technology,
Navi Mumbai
University of Mumbai, India.

ABSTRACT
The Rapid progression in modern internet technology has provoked people to communicate and express by sharing images, video, and other forms of online media. The better image retrieval techniques is increasing rapidly. Vector Quantization (VQ) is one of the lossy image compression techniques. VQ is more efficient than scalar quantization in terms of distortion. VQ comprises of three stages: Codebook Generation, Image Encoding and Image Decoding. The key component of VQ is the codebook generation. The performance of VQ depends on the quality of the codebook generated.
The performance of five different codebook generation techniques namely the Linde, Buzo, and Gray (LBG), Kekre's Proportionate Error Algorithm (KPE), Kekre's Fast Codebook Generation (KFCG), Kekre Error Vector Rotation algorithm (KEVR) and Kekre's Efficient Fast Algorithm (KEFA) for Vector Quantization have been analyzed. In this paper various global codebook generation algorithms for color images are presented.

General Terms
Pattern recognition, Theory.

Keywords
Vector Quantization , Clustering, CodeVector, MSE, PSNR, Global Codebook.

1. INTRODUCTION
The field of Image compression continues to grow at a rapid pace. In future, the need to store and transmit images will only continue to increase faster than the available capability to process all the data. Even with the rapid growth in computer power and increase in the internet bandwidth, The ability to process and transmit the desired amount of image data continues to be problematic. Moreover, advances in video technology and the corresponding growth in the multimedia market, including high-definition television are creating a demand for new, better and faster image. Compression algorithms. Applications that require image compression are many and varied. The massive amount of data required for images is a primary reason for the development of many image compression techniques. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the images can be sacrificed for the sake of saving a little more bandwidth or storage space. Basically, the digital images can be divided into the spatial domain and frequency domain. Before compressing the image data of the frequency domain, we must transform the image data of the spatial domain into the frequency domain, for example, using Discrete Cosine Transform or Hadamard Transform. Vector quantization [1] is one of the most used ways to compress the image data in the spatial or frequency domain. Compression technique which exploits the correlation of the neighboring samples. According to the rate-distortion theorem, better performance is achievable by VQ than scalar quantization.

In the present paper we study the codebook generation in vector quantization (VQ). For codebook generation, two optimizations are essential, first is the Partition optimization and the second is the codebook centroid optimization. The aim is to find a set of representative vectors (called codevectors, or codebook) for a given training set by minimizing the average distortion between the training set and the codebook. It is assumed that the vectors are mapped to their nearest representative in the codebook in respect to a distortion function.

Different types of VQ, such as classified VQ [2], address VQ[2], finite state VQ[2] side match VQ[2], mean-removed classified VQ[3], [4], and predictive classified VQ[3], have been used for various purposes. VQ has been applied to other applications, such as index compression [5], [7], and inverse half toning [3],[6], [7].The important component of a VQ algorithm is the codebook generation. The most widely used technique for codebook generation is the Linde, Buzo and Gray (LBG) algorithm [8]. With its relatively simple structure and computational complexity, VQ has received much attention in the last decade. The performance of the VQ highly depends on the effectiveness of the codebook.The most widely used Generalized Lloyd Algorithm (GLA) starts with an initial solution, which is iteratively improved using two optimality criteria until a local minimum is reached. A codebook can also be built hierarchically. The iterative splitting algorithm [9], [10] starts with a codebook of size one, which is the centroid of the entire training set. The codebook is then iteratively enlarged by a splitting procedure until it reaches the desired size.
Another hierarchical algorithm, the Pairwise Nearest Neighbor (PNN), uses an opposite, bottom-up approach to generate the codebook. It starts by initializing a codebook. Texture feature extraction using VQ based methods viz. LBG and KEVR are discussed. Research efforts in codebook generation techniques have been concentrated in two directions: one to generate a better codebook that approaches global optimal solution, and the other to reduce the computational complexity.

2. VECTOR QUANTIZATION
The vector quantization is one of the primary image compression techniques while compressing the image data in the spatial or frequency domain. It is so called the Loss Digital Image Compression in which the distortions have

occurred during the restored image procedure and the restored image will differ from the original one after compression procedure. 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 and given as $Q:R^k \rightarrow C$, where the codeword $C=\{Y_1, Y_2, Y_3, \ldots, Y_N\}$, and $Y_i \in R^k$. The set C is called the codebook and Y_i, $1 \le i \le N$ are called codewords. In VQ structure, the process of set of training $\{x_i, i=1,2,\ldots,n\}$ that corresponding to the set of codeword $\{y_j, j=1,2,\ldots,m\}$ is called VQ encoding, and the set of codeword is called codebook. The goal of VQ encoding is to generate one index of training sample among codebooks which represents the smallest Euclidean distance named as E[d(Xi,Yj)] between codeword and training vector.

Two error metrics that are used to compare the various image compression techniques are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error. The mathematical formulae for the two are (1) and (2) respectively.

$$MSE = \sum_{y=1}^{M}\sum_{x=1}^{N}\left[I(x,y) - I'(x,y)\right]^2$$

…(1)

$$PSNR = 20 * \log_{10}\left(\frac{255}{sqrt(MSE)}\right)$$

….(2)

where I(x,y) is the original image, I'(x,y) is the approximated version (the reconstructed image) and M,N are the dimensions of the images. The efficiency of VQ mainly depends on the quality of the codebook. The performance of the codebook generation techniques is measured in terms of time taken to generate the codebook and the quality of the reconstructed images using the codebooks.

A lower value for MSE means lesser error, and as seen from the above equations, PSNR is the inverse of MSE and a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. The signal is the original image and the noise is the error in reconstruction. Hence if we find a compression scheme having a lower MSE (and a high PSNR), we can recognize that it is a better one. The MSE can find out the nearest or most representative vector of codebook. A good quality of restoration is adopted by the Peak Signal-to-Noise Ratio (PSNR).

The basic process flow of VQ is Codebook Generator, Vector Encoder and Vector Decoder as follows:

Step 1: Codebook generating:

1. Depend on the selected way to generate codebooks.

Step 2: Encoding:

1. Cut the original image into regular size image blocks.
2. Calculate and find out the codeword most closing to the image block of codebook. After that, record its index.
3. Output the indexes to a file.

Step 3: Decoding:

1. Make the recorded indices correspond to its codebook, and take out the code words.
2. Restore the codeword to the position of original training sample one by one until the original image shows up.

A variety of encoding methods are available in literature: Partial Distortion search (PDS) [11], nearest neighbor search algorithm based on orthonormal transform (OTNNS) [12].Partial Distortion Elimination (PDE) [13], Kekre's fast search algorithms [14], [15], etc., are classified as partial search methods. All these algorithms minimize the computational cost needed for VQ encoding keeping the image quality close to Exhaustive search algorithm.

3. SIMILARITY MEASURES

Many Current Retrieval systems take a simple approach by using typically norm-based distances (e.g., Euclidian distance [16]) on the extracted feature set as a similarity function. The main premise behind these CBIR systems is that given a "good set" of features extracted from the images in the database, then for two images to be "similar" their extracted features have to be "close" to each other. Most commonly Euclidian distance and correlation coefficient are used as similarity measure in CBIR.

Correlation coefficient measures the cosine of the angle between two vectors and varies between 0 to 1. When it is 1 both the vectors are aligned but their magnitude may not be same. In contrast to this Euclidian measure gives the distance between the vectors, when it is 0 not only the vectors are aligned but their magnitude is also same. Here we have preferred Euclidian distance as a similarity measure. The direct Euclidian distance between an image P and query image Q can be given as below.

$$ED = \sqrt{\sum_{i=1}^{n}(V_{pi} - V_{qi})^2}$$

where, Vpi and Vqi be the feature vectors of image P and Query image Q respectively with size 'n'.

4. CODEBOOK GENERATION USING VECTOR QUANTIZATION

4.1 Linde Buzo Gray (LBG) Algorithm [17]

In this method, centroid of the entire training set is computed first. A constant error is added to the centroid. By adding the constant error to and subtracting the constant error from the centroid respectively, two different vectors v1 and v2 are obtained. Then two clusters are formed by grouping the nearest vectors of v1 and v2 using the minimum distance method. The centroids of these two clusters are computed and again the constant error is added to the centroids to get further codevectors. These steps are repeated until the desired size of codebook is generated.

The shortcoming of this algorithm is that the cluster elongation is +135° to horizontal axis in two dimensional cases resulting in inefficient clustering. It presents high MSE. Relative figure for LBG in 2-D case as shown,

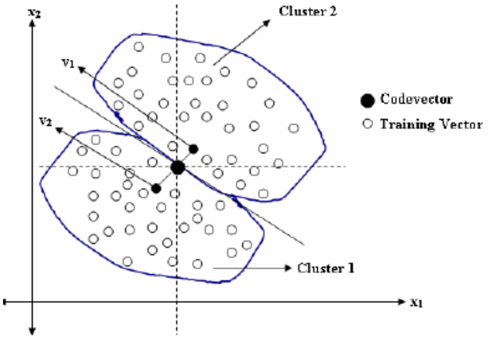

Figure 1. LBG for 2-Dimensional Case

4.2 Kekre's Proportionate Error Algorithm (KPE) [18] :

In this method, a proportionate error is added to the centroids of the clusters. The error value is decided based on the components of the centroid. The minimum cj of the components of the codevector as
{c1, c2, c3, …, ck} is computed. The error value ei is calculated as:

if ci/cj <=10 then
assign ei=ci/cj
else
assign ei=10

where,ci is the individual component of the codevector and cj is the minimum of the components. ei is the component of the error vector{e1, e2,..., ek}. Every time when the new clusters are formed, the error vector is generated and is added to the codevectors to form the new codevectors. This two algorithm takes more time, since Euclidean distance computation is required.
While adding proportionate error a safe guard is also introduced so that neither v1 nor v2 go beyond the training vector space eliminating the disadvantage of the LBG.

4.3 Kekre's Efficient Fast Algorithm (KEFA) [20] :

In this method, the image is divided into small blocks of the size
4 x 4 pixels and the blocks are converted to the codevectors of dimension k, where k = 4 x 4. Giving a matrix T of size N x k consisting of N number of image training vectors of dimension k, each row of the matrix is the image training vector of dimension k. The training vectors are sorted with respect to the first column of the matrix T and the entire matrix is considered as one single cluster. The median of the matrix T is chosen as codevector for the codebook. The matrix is then divided into to two equal parts and each of the part is again sorted based on the second column of the matrix and we obtain two clusters both consisting of equal number of vectors. The median of both the parts are picked up as codevectors for the codebook.Now the size of the codebook is two and two other clusters are formed. These clusters are again divided into four other parts and the median of the four parts are selected to give further codevectors. The above steps are repeated until a codebook of desired size is reached. This algorithm takes very less time compared to the above two algorithms, since Euclidean distance computation is not required.

4.4 Kekre's Efficient Vector Rotation (KEVR)

The disadvantage of LBG is overcome in Kekre"s Error Vector Rotation (KEVR) algorithm. To generate the codebook, the image is first divided into fixed size blocks, each forming a training vector **Xi = (xi1, xi2, ……., xik)**. The set of training vectors is a training set. This training set is initial cluster. The clustering algorithms like LBG and KEVR are then applied on this initial cluster to generate the Codebook of desired size. LBG is standard VQ codebook generation algorithm.
The KEVR algorithms for codebook generation are discussed. In Kekre Error Vector Rotation algorithm (KEVR) algorithm two vectors v1 & v2 are generated by adding error vector to the codevector. Euclidean distances of all the training vectors are computed with vectors v1 & v2 and two clusters are formed based on closest of v1 or v2. The codevectors of the two clusters are computed and then both clusters are splitted by adding and subtracting error vector rotated in k dimensional space at different angle to both the codevector.

This modus operandi is repeated for every cluster and every time to split the clusters error ei is added and subtracted from the codevector and two vectors v1 and v2 is generated. Error vector ei is the ith row of the error matrix of dimension k. The error vectors matrix E is given in equation 1.

$$E = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ e_k \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & \cdots & -1 & 1 & 1 \\ \cdots & & & & & & & \\ \cdots & & & & & & & \\ \cdot & & & & & & & \\ \cdot & & & & & & & \\ \cdots & & & & & & & \end{bmatrix} \qquad (1)$$

These error vector sequences have been obtained by taking binary representation of numbers starting from 0 to k-1 and replacing zeros by 1'S and ones by -1'S. This algorithm takes very less time, since Euclidean distance computation is not required.It is 99.98% faster than LBG & KPE. It gives the better PSNR.

4.5 Kekre's Fast Codebook Generation (KFCG) [19] :

Codebook Generation (KFCG) algorithm . Image is divided into the windows of size 2x2 pixels (each pixel consisting of red, green and blue components). These are put in a row to get 12 values per vector.Collection of these vectors is a training set (initial cluster).Compute centroid (codevector) of the cluster. Compare the first element of the training vector with the first element of the codevector and split the above cluster into two. Compute the centroids of both the clusters obtained. Split both the clusters by comparing second element of training vectors with the second element of the codevectors. Repeat the process till we obtain codebook of desired size.

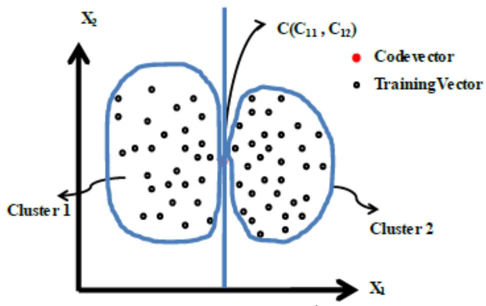

Figure 2a. 1st Iteration

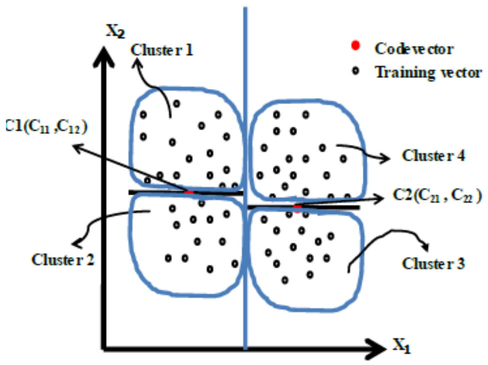

Figure 2b. 2nd Iteration

Figure 2.KFCG algorithm for 2-D case

5. CONCLUSION

Today we are living in the information age, where images have giant share in this information. Clustering technique is adopted for optimizing the initial codebook that is generated by any one of the methods. More précised retrieval techniques are needed to access the large image archives being generated, for finding relatively similar images. In this paper image retrieval technique is discussed using vector quantization which is a popular technique for data compression. For VQ technique we have used KFCG algorithm to generate codebook which is very fast as it does not involve any Euclidian distance computation. This technique for CBIR has far less complexity. On an average, We have discussed five different techniques for generating the initial codebook. In LBG constant error is added every time to split the clusters, which results in cluster formation in one direction only. The cluster elongation in LBG is 135˚ in 2-dimensional case. Due to this reason clustering in LBG is inefficient resulting in high MSE. To overcome this drawback of LBG modification to it is introduced by adding proportionate error to change the cluster orientation in KPE.Thus KFCG splitting the cluster comparatively faster. This has resulted in improving the clustering and reducing the image degradation in reconstructed image considerably.

6. REFRENCES

[1] Gray, R. M., 1984, “Vector quantization,” *IEEE ASSP Magazine*, Vol. 1, pp. 4-29

[2] Jim Z.C. Lai, Yi-Ching Liaw, and Julie Liu, “A fast VQ codebook generation algorithm using codeword displacement,” Pattern Recogn.vol. 41, no. 1, pp 315–319, 2008.

[3] J. Foster, R.M. Gray, M.O. Dunham, “Finite state vector quantization for waveform coding,” IEEE Trans. Inf. Theory vol. 31, No. 3, pp. 348–359, 1985.

[4] J. Z. C. Lai, Y.C. Liaw, W. Lo, “Artifact reduction of JPEG coded images using mean-removed classified vector quantization,” Signal Process. vol. 82, No. 10, pp. 1375–1388, 2002.

[5] C. H. Hsieh, J.C. Tsai, “Lossless compression of VQ index with search order coding,” IEEE Trans. Image Process. vol. 5, No. 11, pp. 1579–1582, 1996.

[6] J. C. Lai, J.Y. Yen, “Inverse error-diffusion using classified vector quantization,” IEEE Trans. Image Process. vol. 7, No. 12, pp. 1753–1758, 1998.

using Vector Quantization,” International Journal of Engineering and Technology, vol.1, No.1, pp.: 67-77, September 2008.

[7] P.C. Chang, C.S. Yu, T.H. Lee, “Hybrid LMS-MMSE inverse halftoning technique,” IEEE Trans. Image Process. vol. 10, No. 1, pp.95–103, 2001.

[8] Y.Linde, A.Buzo and R.M.Gray, “An Algorithm for Vector Quantizer Design,” IEEE Trans. Communication, Vol. 28, pp. 84-95,Jan. 1980.

[9] X.Wu and K.Zhang, “A Better Tree-Structured Vector Qunatizer,” IEEE Proc., Data Compression Conference, Snowbird, UT, 1991, pp. 392-401.

[10] P.Franti, T.Kaukoranta, and O.Nevalainen, “On the Splitting Method for VQ Codebook Generation,” Opt. Eng., vol 36, pp.3043-3051, Nov.1997.588

[11] C. D. Bei and R. M. Gray.: ‘An improvement of the minimum distortion encoding algorithm for vector quantization’, IEEE Trans Commun.,vol. 33, No. 10, pp. 1132–1133, Oct. 1985.

[12] Z. Li, and Z.- M. Lu. : ‘Fast codevector search scheme for 3D mesh model vector quantization’, Electron. Lett., vol. 44, No. 2, pp. 104-105, Jan 2008.

[13] C. Bei, R. M. Gray, ‘‘An improvement of the minimum distortion encoding algorithm for vector quantization’’, IEEE Trans. Commun.,Vol. 33, no. 10, pp. 1132–1133, Oct 1985.

[14] H. B. Kekre, Tanuja K. Sarode, “Fast Codevector Search Algorithm for 3-D Vector Quantized Codebook”, WASET International Journal of cal Computer Information Science and Engineering (IJCISE),Volume 2, No. 4, pp. 235-239, Fall 2008. available:http://www.waset.org/ijcise.

[15] H. B. Kekre, Tanuja K. Sarode,”Fast Codebook Search Algorithm for Vector Quantization using Sorting Technique”, ACM International Conference on Advances in Computing, Communication and Control(ICAC3-2009), 23-24 Jan 2009, Fr. Conceicao Rodrigous College of Engg., Mumbai. Available on online ACM portal.

[16] H.B.Kekre, Sudeep D. Thepade, “Boosting Block Truncation Coding using Kekre’s LUV Color Space for Image Retrieval”, WASET Int. Journal of Electrical, Computer and System Engineering (IJECSE), Vol.2, Num.3, Summer 2008. online at www.waset.org/ijecse/v2/v2-3-23.pdf

[17] S. C. Lo, H. P. Chan, J. S. Lin, H. Li, M. T. Freedman, and S. K. M un, “Artificial convolution n eural network for medical image p attern recognition,” Neural Networks, vol.8, no. 7/8, pp. 1201–1214, 1995. H.B.Kekre, Tanuja K. Sarode, “New Fast Imp roved Clustering Algorithm for Codebook Generation for Vector Quantization”, International Conferen ce on Engineering Technologies and Applications in Engineering, Technology and Sciences, Computer Science Department, Saurashtra University, Rajkot, Gujarat. (India), Amoghsiddhi Education Society, Sangli, Maharashtra (India) , 13th – 14th January 2008.

[18] H. B. Kekre, Tanuja K. Sarode, “Fast Codebook Generation Algorithm for Color Images using Vector Quantization,” International Journal of Comp uter Science and Information Technology, Vol. 1, No. 1, pp.: 7-12, Jan 2009.

[19] H. B. Kekre, Tanuja K. Sarode, “New Fast Improved Codebook Generation Algorithm for Color Images