

An Improved Image Compression method using LBG with DCT

Pallavi N. Save

P.G. student, Electronics and Telecommunication
Department
D.J. Sanghvi College of Engineering
Mumbai, India

Vishakha Kelkar

Assistant Professor, Electronics and Telecommunication
Department
D.J. Sanghvi College of Engineering
Mumbai, India

Abstract— This paper presents new algorithm based on discrete cosine transform and LBG algorithm for vector quantization for image compression. Vector quantization is mainly divided into three parts i.e Encoding process, Codebook design, Decoding process. In vector quantization generation of codebook is important so that the distortion between the original image and the reconstructed image is minimum. It is observed that the performance of new algorithm LBG using DCT is better than LBG. Performance is measured using PSNR and MSE.

Keywords— VQ, LBG, DCT

I. INTRODUCTION

Vector quantization (VQ) is an effective method of data compression. The main objective of image compression is to reduce redundant data of the image in order to store or transmit data efficiently. Image compression can be classified as lossy or lossless. Vector Quantization (VQ) is an efficient and simple approach for data compression. Vector quantization is often used when high compression ratios are required. Lossy compression techniques are based on quantization. Quantization is irreversible process. The important Lossy compression techniques are Scalar quantization and Vector quantization. Quantizing each pixel separately is called scalar quantization. Quantizing group of pixels together is called Vector quantization.

Vector quantization is the process of dividing an image into vectors or blocks and each vector is mapped to the codeword's of a codebook to find its reconstructed vector. Each of these vectors is then approximated and replaced by the index of one of the elements of the codebook. There are three major procedures in Vector Quantization, namely codebook generation, encoding procedure and decoding procedure. The goal of the codebook design procedure is to design the desired codebook that is to be used in the image encoding/decoding procedures.

In VQ an original image of size $N \times N$ is divided into blocks of size $n \times n$. Therefore total numbers of blocks are $n_b = (N/n) \times (N/n)$. n_b is known as input vectors as $X = (X_1, X_2, \dots, X_{n_b})$, $C = (C_1, C_2, \dots, C_{n_c})$ is known as codebook in which n_c indicates the total number of codevectors. Vector quantization is the process of mapping each input vector to the one of the codevector from the codebook. Mapping is based

upon the minimum distance criterion i.e by finding out Euclidian distance. The closest codeword in the codebook for each image vector is to be determined. The compressed codes of VQ are the indices of the closest codewords in the codebook for all the image blocks.

One of the main issues of VQ encoding is to reduce the computational complexity from searching the best matched codeword in order to reduce the search time. Many efficient algorithms have been developed to reduce the computation load.

In LBG [1] algorithm, the square of Euclidean distance between each input vector and each codeword in the codebook has to be calculated to find the closest codeword. This is a very time-consuming process when the number of vectors in the training set is large. Hence the computational complexity of the LBG algorithm is high. To decrease the computation cost of the LBG algorithm, some fast algorithms for codebook searching have been proposed, such as the partial distortion search (PDS) [2] and the mean-ordered partial codebook search (MPS)[3].

In Pairwise Nearest Neighbor[4] method first codebook of size N is initialized, where each input vector is considered as its own codevector. In each step two codevectors are merged and this process is repeated till codebook reduces to size M . Main drawback of PNN is its slowness because implementation require a lot time. The Pairwise Nearest Neighbor (PNN) algorithm is an alternative method to the LBG algorithm for generating codebooks.

This paper describes an improved algorithm which uses discrete cosine transform and LBG algorithm to form a codebook. Proper selection of the codebook in vector quantization reduces the distortion in the reconstructed image.

The paper is organized as follows: Section II gives methodology. Section III gives results and section IV gives the concluding remarks.

II. METHODOLOGY

A. LBG[1]

LBG algorithm for codebook design is an iterative procedure. The procedure for a codebook generation is given as follows:

1: First step is to generate random codebook CB_0 .

2: $i = 0$.

3: For each input vector perform following process.

- Compute the Euclidean distances between the training vector and the codewords in CB_i . The Euclidean distance is defined as

$$d(X,C) = \sqrt{\sum_{t=1}^k (X_t - C_t)^2} \quad (1)$$

- Search the nearest codeword from CB_i .

4: Partition the codebook into N cells.

5: Compute the centroid to obtain the new codebook CB_{i+1} .

6: Compute the average distortion for CB_{i+1} . If it is changed by a small amount since the last iteration, and the procedure stops.

Otherwise, $i = i + 1$ and go to Step 3.

LBG is an easy and rapid algorithm. In LBG algorithm the resulted codebook quality depends on the initial codebook. If the initial codebook is not proper then it produces a distortion in the reconstructed image.

This algorithm generates very large codebook. It requires large storage space for codebook. It takes a long time for the generation of the codebook. A complete design of this algorithm requires a large number of computations. Convergence time is very large.

B. LBG with DCT[5]

This algorithm uses a DCT (Discrete cosine Transform) and LBG to construct a codebook. In this the input image is divided into blocks or subimages and then DCT coefficients of blocks are calculated and quantized. Next step is generate a codebook using LBG in DCT domain. Applying LBG algorithm in the frequency domain reduces distortion in the reconstructed image. For each input vector a codeword is selected from the codebook. Index metrics indicates the corresponding block codeword. By using this algorithm improvement in the PSNR values as well as higher compression ratio is achieved.

Algorithm

- 1) The input image is subdivided into subblocks or subimages.
- 2) Next step is to shift intensity value of each pixel by $-2^7 = -128$

- 3) Then we apply DCT transform on each subimage or block of input image.
- 4) DCT transform applied on each subblocks.
- 5) Convert 2D matrix into 1D matrix by zigzag ordering pattern. DCT coefficients are then quantized. Nonzero coefficients are concentrated in the first elements of the input vector, so we use few first ones and ignore the rest, which increases the compression rate.
- 6) Find the maximum absolute value of the remained DCT coefficients (m) and change the coefficient values by multiplying each coefficient by $63/m$.
- 7) We use the LBG algorithm to generate the codebook from the input vectors in the DCT domain. The initial codewords are selected from the input vectors obtained in step 6
- 8) In the reconstruction of image we compute the inverse DCT of the each subimage.

C. Improved LBG with DCT

In LBG with DCT algorithm, high frequency components are quantized so more distortion in those blocks, and this is overcome by variance criterion. In that variance of each subblocks is calculated. If the variance of each subblock is greater than the average of variance then dct coefficients of input image will not quantized. If the variance of each subblock is less than the average of variance then dct coefficients of input image will be quantized. This improved LBG with DCT improves the PSNR value as well as image quality as compared with the LBG with DCT.

III. RESULTS AND DISCUSSION

We use the peak signal to noise ratio (PSNR) to evaluate the quality of the reconstructed images.

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \quad (2)$$

Mean square error (MSE) is given as

$$MSE = \left(\frac{1}{N \times N} \right) \sum_{i=1}^N \sum_{j=1}^N (X_{ij} - Y_{ij})^2 \quad (3)$$

Where X_{ij} and Y_{ij} represents the pixel intensity at the position (i,j) of the original and the reconstructed images respectively.

In the experiments, we have used Lena, Cameraman, and house as the test images. The images are of size 256×256 pixels. We use different codebook sizes of 128, 256, 512 and 1024 to evaluate the performance of reconstructed image.

Table 1 shows the MSE and PSNR for Lena with different codebook size and Fig 1 shows LBG algorithm results for the different codebook size. We can see that as the codebook size increases from 128 to 1024 MSE between reconstructed image and original image decreases and PSNR increases.

Table 1. MSE and PSNR using LBG for lena image with different codebook size

Codebook Size	MSE	PSNR
128	2982.4368	13.3851
256	2754.5808	13.7302
512	2737.7713	13.7568
1024	2618.4109	13.9504

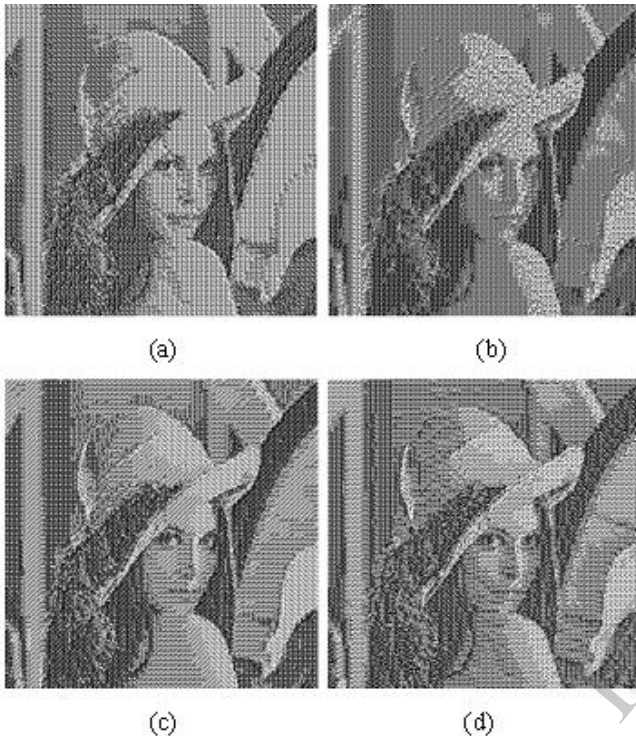


Fig.1 Implementation of LBG algorithm for different Codebook Size (a) codebook size=128,(b) codebook size =256,(c) codebook size =512,(d) codebook size =1024

Fig.2 shows the results for Lena image using LBG with DCT for different codebook size. Table 2 gives the MSE and PSNR values for Lena image with different codebook size. With increase in codebook size from 128 to 1024, Mean square error reduces and PSNR increases. Compression rate is high as all the DCT coefficients are quantized, and only nonzero values are used for codebook.

Table 2. MSE and PSNR using DCT with LBG for lena image with different codebook size

Codebook Size	MSE	PSNR
128	1096.5497	17.7305
256	761.2808	19.3154
512	736.3866	19.4597
1024	511.0909	21.0458

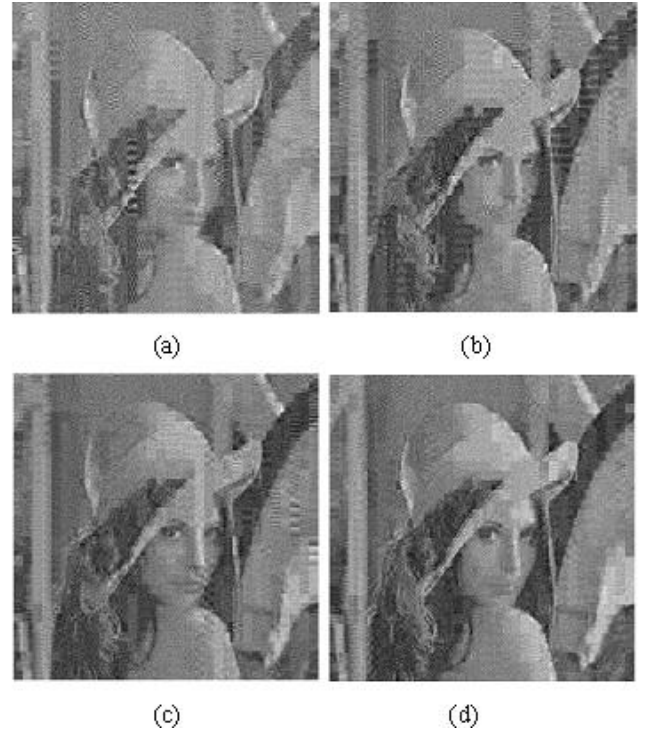


Fig.2 Results using DCT with LBG algorithm for different Codebook Size (a) codebook size=128,(b) codebook size =256,(c) codebook size =512,(d) codebook size =1024

Table 3 gives the MSE and PSNR values for Lena image with different codebook size. Fig.3 shows the results for Lena image using an improved LBG with DCT for different codebook size. We can see that with the increase in codebook size from 128 to 1024, Mean square error (MSE) between original image and reconstructed image is decreases, and PSNR is increases. With this improved LBG with DCT variance condition will decide the how many blocks should get quantized, which reduces the compression rate but image quality is better compared with the LBG and LBG with DCT.

Table 3. MSE and PSNR using Improved DCT with LBG for lena image with different codebook size

Codebook Size	MSE	PSNR
128	245.7108	24.2266
256	168.7501	25.8584
512	122.6438	27.2443
1024	84.7698	28.8484



Fig.3 Results using Improved DCT with LBG algorithm for different Codebook Size (a) codebook size=128,(b) codebook size =256,(c) codebook size =512,(d) codebook size =1024

Fig.5 shows the overall comparison of PSNR with different codebook size from 128 to 1024 for LBG, LBG with DCT and an Improved LBG with DCT. We can see that with an improved LBG with DCT shows higher PSNR values as compared with LBG and LBG with DCT.

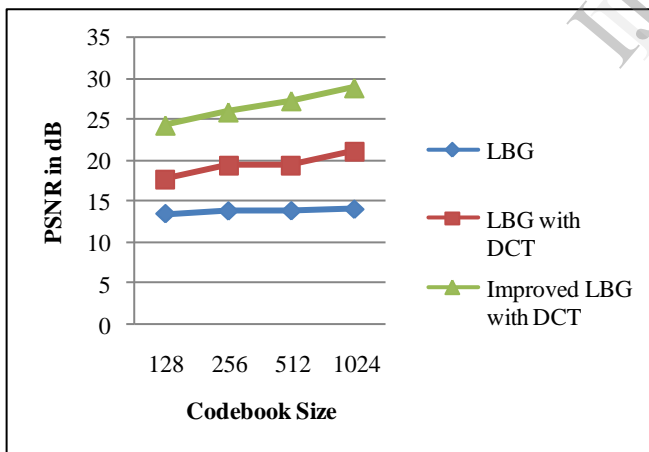


Fig 5: Comparison of PSNR for LBG, LBG with DCT and An Improved LBG with DCT

Fig.6 shows the overall comparison of MSE between the reconstructed image and original image with different codebook size from 128 to 1024 for LBG, LBG with DCT and an Improved LBG with DCT. Comparison shows that with an improved LBG with DCT, MSE decreases with increase in codebook size.

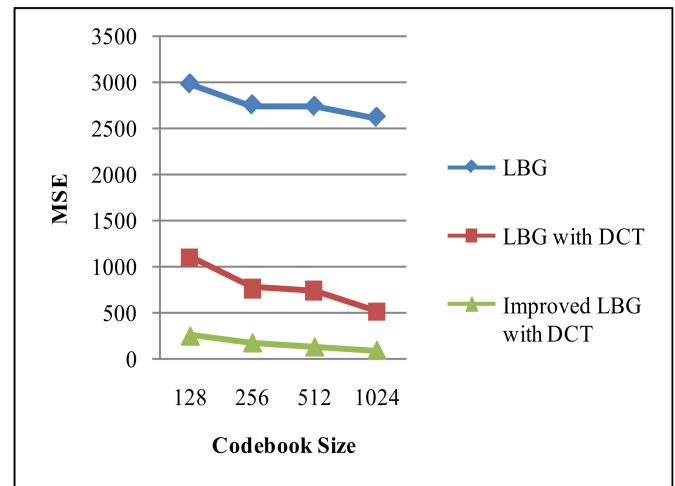


Fig 5: Comparison of MSE for LBG, LBG with DCT and An Improved LBG with DCT

IV. CONCLUSION

An experimental result shows that using Improved LBG algorithm in DCT domain achieves higher PSNR values compared to the LBG algorithm. An improvement in PSNR value also increases as the codebook size increases.

FUTURE WORK

Further the performance of an Improved LBG with DCT can be increases by implementing with Fast search algorithms which reduces the searching time for codevector from the codebook. It reduces the average number of distance calculations.

REFERENCES

- [1] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun., vol. -28, no. 1, pp. 84-95, 1980.
- [2] C.D. Bei &R. Gray, An improvement of the minimum distortion encoding algorithm for vector quantization, IEEE Transactions on Communications, 33 (10), 1985, 1132-1133.
- [3] S. W. Ra and J. K. Kim, "A Fast Mean-Distance-Ordered Partial Codebook Search Algorithm for Image Vector Quantization", IEEE Trans. Circuits and Systems - II: Analog and Digital Signal Processing, vol. 40, no. 9, pp. 576{579, 1993.
- [4] W. H. Equitz, "A new vector quantization clustering algorithm," IEEE Trans. Acoust. Speech Signal Process. 37~10, 1568-1575 1989.
- [5] Mahmood Shabanifard and Mahrokh G. Shayesteh, "A New Image Compression Method Based on LBG Algorithm in DCT Domain" , IEEE transactions, 2011
- [6] N.M. Nasrabadi, Y. Feng, Image compression using address vector quantization, IEEE Trans. Commun. vol. 38 No. 12, pp. 2166-2173, 1990.
- [7] C. M. Huang and R. W. Harris, A Comparison of Several Vector Quantization Codebook generation Approaches, IEEE Trans. Image Processing, vol. 2, no. 1, pp. 108-112, 1993.
- [8] T. C. Lin and P. T. Yu, Centroid Neural Network Adaptive Resonance Theory for Vector Quantization, Signal Processing, vol. 83, pp. 649{654, 2003.
- [9] C.M. Huang, Q. Bi, G.S. Stiles, & R.W. Harris, Fast full search equivalent encoding algorithms for image compression using vector quantization, IEEE Transactions on Image Processing, 1 (3), 1992, 413-416.

- [10] QIAN S.E.: 'Fast vector quantisation algorithms based on nearest partition set search', IEEE Trans. Image Process., 2006, 15, (8), pp. 2422–2430.
- [11] LEE C.H., CHEN L.H.: 'Fast closest codeword search algorithm for vector quantisation', IEE Proc Vis. Image Signal Process., 1994, 141, (3), pp. 143–148
- [12] BAEK S.J., JEON B.K., SUNG K.M.: 'A fast encoding algorithm for vector quantisation', IEEE Signal Process. Lett., 1997,4, (12), pp. 325–327

IJERT