An Improved Image Compression Method using Vector Quantization for Color Images

Pallavi N. Save
P.G. Student
Department of Electronics & Telecommunication Engineering
D.J. Sanghvi College of Engineering

Vishakha Kelkar
Assistant Professor
Department of Electronics & Telecommunication Engineering
D.J. Sanghvi College of Engineering

ABSTRACT
This paper presents new algorithm based on discrete cosine transform and LBG algorithm using vector quantization for image compression of color images. 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. In this RGB component of color image are converted to YCbCr before DCT transform is applied. Y is luminance component; Cb and Cr are chrominance components of the image. This paper compares three different algorithms for vector quantization of color images. 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.

1. INTRODUCTION
With the advanced development in the technologies Storage and transmission of digital image in multimedia systems is major problem, because the amount of data required to present images at an acceptable level of quality is extremely large. High quality image data requires large amount of storage space and transmission bandwidth. To solve this problem one solution is to compress the information so that the storage space and transmission time can be reduced [1]. Color image usually contains a lot of data redundancy and requires a large amount of storage space. In order to lower the transmission and storage cost, image compression is done. Quantization is the process of reducing the number of possible values of a quantity, thereby reducing the number of bits needed to represent it. 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 [1]. Vector Quantization (VQ) is an efficient approach for data compression. Vector quantization is often used when high compression ratios are required. Vector quantization is the process of dividing an image into non-overlapping blocks of uniform size known as input vectors and each vector is mapped to the codeword's of a codebook to find its reconstructed vector. 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 vector quantization generation of codebook is important because the distortion between the original image and the reconstructed image is minimum.In Vector Quantization an original image of size N X N is divided into blocks of size n x n. Therefore total numbers of blocks are N_b = (N/n) X (N/n). N_b is known as input vectors and represented as X=(X_1, X_{2} , ... X_{nb}) . C= (C_1, C_2,……C_{Nc}) 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 one of the codevector from the codebook. Mapping is based upon the minimum distance criterion i.e by finding out Euclidian distance. The closest codevector in the codebook for each image vector or input vector is to be determined. Replace the vector by the index in codebook. The compressed codes of VQ are the indices of the closest codevectors in the codebook for all the image blocks. Compression is achieved because the indices of the closest codevectors in the codebook are sent to the decoder instead of the image blocks themselves.In this paper vector quantization is applied on the color images. Most color images are recorded in RGB model, which is the most well known color model. However, RGB model is not suited for image compression purpose. For compression, a luminance-chrominance representation is considered superior to the RGB representation. Therefore, RGB images are transformed to one of the luminance-chrominance models, and then transform back to RGB model because displays are most often provided output image with direct RGB model. In this paper RGB component of color image are converted to YCbCr before DCT transform is applied. Y is luminance component; Cb and Cr are chrominance components of the image. The luminance component represents the intensity of the image and look likes a gray scale version. The chrominance components represent the color information in the image.Sadashivappa[3] proposed method for Color image compression using an SPIHT algorithm. The SPIHT algorithm is applied for luminance (Y) and chrominance (Cb,Cr) part of RGB to YCbCr transformed image. Reconstructed image is verified using image quality and PSNR. Fouzi Douak proposed method [4], in which After a preprocessing step (mean removing and RGB to YCbCr transformation), the DCT transform is applied and followed by an iterative phase (using the bisection method) including the threshold, the quantization, dequantization, the inverse DCT, YCbCr to RGB transform and the mean recovering.In LBG [2] algorithm, 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.This paper describes an improved algorithm for the image compression of color images 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.

2. METHODOLOGY

2.1 LBG [2]

1) Convert RGB image into YCbCr color format.
2) Take only Y component of YCbCr image, which will be the input image.
3) Input image of size N X N is divided into subblocks of size n x n.
4) Define the size of codebook and randomly select M vectors from the input vectors.

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

Search the nearest codevector from C and then add the input vector into corresponding cluster of the closest codevector found.

5) For each codevector in the current codebook C, find the centroid of its associated cluster and take the centroid as a new codevector for the next iteration. Repeat step 2 and 3 till codevector don't change or change is small.
6) Measure the performance parameters PSNR and MSE of the reconstructed RGB image.

The above algorithm is executed with the different codebook size of 128, 512 and 1024 for the analysis of the parameters and effect of codebook size on the reconstructed image.

2.2 LBG algorithm with DCT [5]

Algorithm 2 based on LBG implemented with DCT.

1) Convert RGB image into YCbCr color format.
2) Take only Y component of YCbCr image, which will be the input image.
3) The input image of size N X N is subdivided into subblocks.
4) Next step is to shift intensity value of each pixel by $-2^7=$ -128.
5) Then apply DCT transform on each block of input image.
6) 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 using this few nonzero elements and ignoring the rest, which increases the compression rate at the cost of image quality.
7) Find the maximum absolute value (m) of the remained DCT coefficients and change the coefficient values by multiplying each coefficient by 63/m. This will embed each coefficient in the interval of [-63, 63].
8) The LBG algorithm is used to generate the codebook from the input vectors in the DCT domain. The initial codewords are selected from the input vectors obtained in step 5.
9) Perform Inverse zigzag.
10) In the reconstruction of image we compute the inverse DCT of the each subblock.
11) Perform the inverse level shift.
12) Measure the performance parameters PSNR and MSE of the reconstructed RGB image.
13) The above algorithm is executed with the different codebook size of 128, 512 and 1024 for the analysis of the parameters and effect of codebook size on the reconstructed image.

2.3 An Improved LBG with DCT [6]

In LBG with DCT algorithm, coefficients that correspond to high frequencies are quantized. Because these high frequency components are quantized distortion is introduced. To improve the overall quality of the reconstructed image variance criterion is used to select the block where quantization will be more. This improved LBG with DCT improves the PSNR value as well as image quality as compared with the LBG algorithm using DCT**.**

In an Improved LBG with DCT method variance of each subblock 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.

The above algorithm is executed with the different codebook size of 128, 512 and 1024 for the analysis of the parameters and effect of codebook size on the reconstructed image.

2.4 Sampling in an Improved LBG with DCT

Chroma subsampling is the method of encoding images by implementing less resolution for chrominance information than luminance information. In YCbCr color format, the Cr Cb components are filtered and then sub-sampled at ½ or ¼ the rate of the Y component, with visually same image quality. There are several YCbCr format such as 4:4:4, 4:2:2 and 4:1:1.The sampling format implies that sampling rate of Cb and Cr are one half of that of Y [1][8].

This method uses 4:4:2, 4:2:4 and 4:2:2 sampling. In 4:4:2 sampling Cr component is sampled to one half of Y and Cb component. In 4:2:4 sampling Cb component is sampled to one half of Y and Cr component. In 4:2:2 sampling both the chrominance components Cb and Cr is sampled to one half of Y component.

The above algorithm is executed with the different codebook size of 128, 512 and 1024 for the analysis of the parameters and effect of codebook size on the reconstructed image.

3. RESULTS AND DISCUSSIONS

Performance parameters such as peak signal to noise ratio (PSNR) and Mean Square error (MSE) are used 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

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

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

Lower value for Mean square error means lesser error and PSNR is the inverse of MSE and a higher the 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.In the experiments, color images of Lena, Cat, and house are the test images. The images are of size 256 x 256 if the image size is not 256 x 256 then next step is to resize the image or to make image such that it will be multiples of 8. The different codebook sizes of 128, 256, 512 and 1024 to evaluate the performance of reconstructed image.LBG algorithm requires initial random codebook, for which codevectors are selected from the input vectors of input image. If the initial codebook is not proper then it produces a distortion in the reconstructed image.

Table 1 shows results for Lena image using LBG algorithm with different codebook size from 128 to 1024 and Figure 1 shows the MSE and PSNR for Lena image with different codebook size. It was observed that as the codebook size increases from 128 to 1024 Mean square error (MSE) between reconstructed image and original image decreases from 3328.2472 for codebook size 128 to 2633.714 for codebook size 1024. PSNR increases to 12.9086 for codebook size 128 to 13.9251 for codebook size 1024.

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

Codebook Size	MSE	PSNR
128	3328.2472	12.9086
256	3213.32	13.0613
512	2697.92	13.8205
1024	2633.714	13.9251

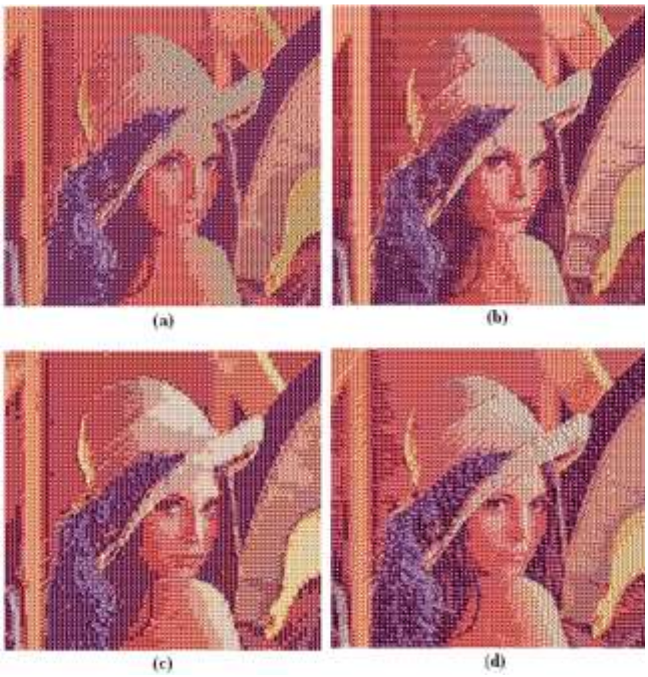

Figure 1 Reconstructed Images using LBG algorithm for lena Image with different codebook size (a) Codebook size=128,(b) Codebook size =256, (c) Codebook size =512, (d) Codebook size =1024

Table 2 shows results for Lena image using LBG with DCT algorithm with different codebook size from 128 to 1024 and Figure 2 shows the MSE and PSNR for Lena image with different codebook size. It was observed that as the codebook size increases from 128 to 1024 Mean square error (MSE) between reconstructed image and original image decreases from 929.7374 for codebook size 128 to 413.7974 for codebook size 1024. PSNR increases to 18.4472 for codebook size 128 to 21.9629 for codebook size 1024.

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

Codebook Size	MSE	PSNR
128	929.7374	18.4472
256	768.8089	19.2726
512	472.4724	21.387
1024	413.7974	21.9629

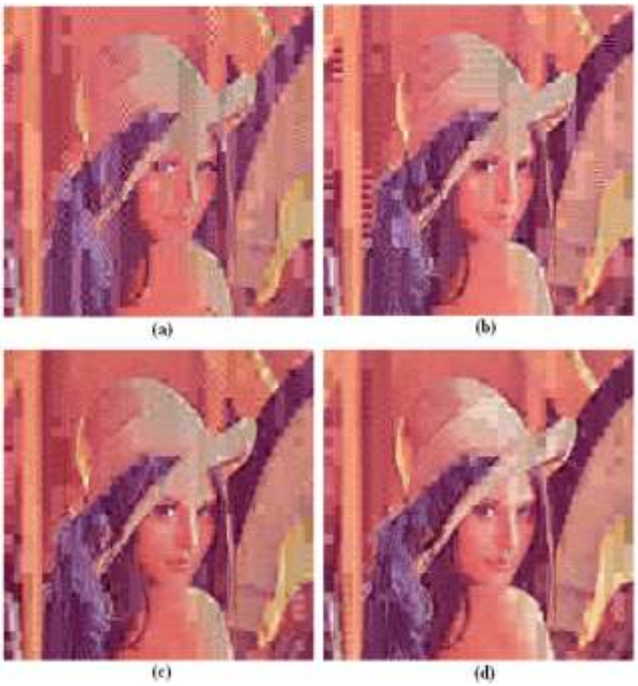

Figure 2 Reconstructed Images using LBG algorithm with DCT for lena Image with different codebook size (a) Codebook size=128,(b) Codebook size =256, (c) Codebook size =512, (d) Codebook size =1024

Figure 3 shows results for Lena image using An Improved LBG with DCT algorithm with different codebook size from 128 to 1024 and Table 3 shows the MSE and PSNR for Lena image with different codebook size. It was observed that as the codebook size increases from 128 to 1024 Mean square error (MSE) between reconstructed image and original image decreases from 3328.2472 for codebook size 128 to 2633.714 for codebook size 1024. PSNR increases to 12.9086 for codebook size 128 to 13.9251 for codebook size 1024.

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

Codebook Size	MSE	PSNR
128	576.4018	20.5236
256	292.5193	23.4693
512	177.687	25.6342
1024	165.9484	25.9484

Figure 3 Reconstructed Images using An Improved LBG algorithm with DCT for lena Image with different codebook size (a) Codebook size=128,(b) Codebook size =256, (c) Codebook size =512, (d) Codebook size =1024

Table 4 shows the MSE and PSNR for Lena image using 4:4:2 sampling in an Improved LBG with DCT with Different codebook Size.

Table 4. MSE and PSNR using 4:4:2 sampling on an Improved LBG with DCT for lena image with different codebook size

Codebook Size	MSE	PSNR
128	576.5299	20.522
256	292.6093	23.4679
512	177.7558	25.6326
1024	165.3404	25.947

Table 5 shows the MSE and PSNR for Lena image using 4:2:4 sampling in an Improved LBG with DCT with Different codebook Size.

Table 5. MSE and PSNR using 4:2:4 sampling on an Improved LBG with DCT for lena image with different codebook size

Codebook Size	MSE	PSNR
128	576.7255	20.5211
256	292.8302	23.4646
512	178.0489	25.6254
1024	165.65	25.9389

Table 6 shows the MSE and PSNR for Lena image using 4:2:2 sampling in an Improved LBG with DCT with Different codebook Size.

Table 6. MSE and PSNR using 4:2:2 sampling on an Improved LBG with DCT for lena image with different codebook size

Codebook Size	MSE	PSNR
128	576.8376	20.5203
256	292.9023	23.4636
512	178.1121	25.6239
1024	165.7016	25.9375

Chrominance components Cb and Cr contain visually less information. Results in the table 4, table 5 & table 6 shows that sampling of these components will not affect the image quality and maintain PSNR as with the only Y component.

Fig.4 shows the overall comparison of MSE of the reconstructed images with different codebook size from 128 to 1024 for LBG, LBG with DCT and an Improved LBG with DCT. It was observed that with an improved LBG with DCT shows higher PSNR values as compared with LBG and LBG with DCT.

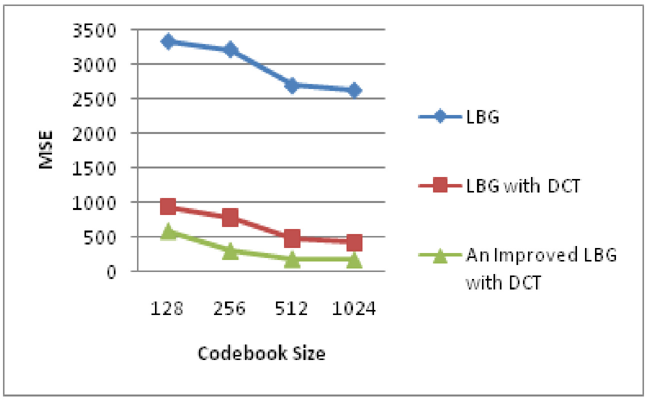

Figure 4: Comparison of MSE for LBG, LBG with DCT and an Improved LBG with DCT

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. It was observed that with an improved LBG with DCT shows higher PSNR values as compared with LBG and LBG with DCT.

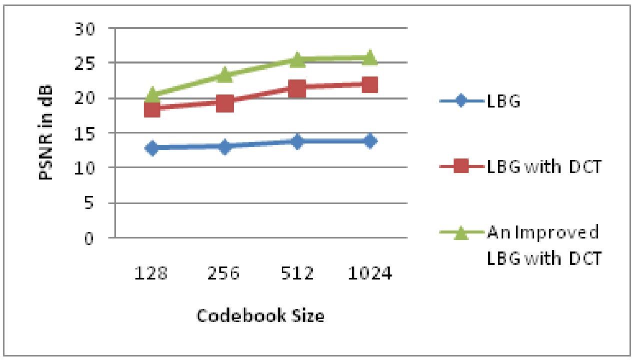

Figure 5: Comparison of PSNR for LBG, LBG with DCT and an Improved LBG with DCT

4. CONCLUSION

An experimental result shows that in LBG algorithm reconstructed image have more Mean square error and less PSNR, performance of LBG algorithm improved using a new codebook design method implemented with DCT. This new method shows improvement in PSNR but overall image quality was degraded. Further the performance of new codebook design method and overall image quality of reconstructed image improved using variance criterion. Variance criterion used to select the blocks where quantization will be more. This improved LBG with DCT improves the PSNR value as well as image quality as compared with the LBG with DCT. An improvement in PSNR value also increases as the codebook size increases at the cost of execution time. Furthermore compression achieved through a sampling of chrominance components. An experimental result shows that in an Improved LBG algorithm using DCT domain with sampling of chrominance components achieves higher PSNR values compared to the LBG algorithm. Quality of image still retained, though chrominance components Cb and Cr are compressed up to 50%. An improvement in PSNR value also increases as the codebook size increases. Drawback of the above algorithm is with the increase codebook size the effective execution time & complexity increases proportionally. To improve this various fast search methods can be incorporated in future.

5. REFERENCES

[1] S Jayaraman, S Esakkirajan, T Veerakumar, Digital Image Processing,2009.

[2] 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.

[3] Sadashivappa M. J., K.V.S Anand Babu, Dr. Srinivas K,"Color Image Compression using SPIHT Algorithm," International Journal of Computer Applications (0975 – 8887),Vol. 16– No.7, February 2011.

[4] Fouzi Douak, Redha Benzid, Nabil Benoudjit "Color image compression algorithm based on the DCT transform combined to an adaptive block scanning," Int. J. Electron. Commun. (AEU), vol. 65, pp. 16–26, 2011.

[5] Mahmood Shabanifard and Mahrokh G. Shayesteh, "A New Image Compression Method Based on LBG Algorithm in DCT Domain" , IEEE transactions, 2011.

[6] Pallavi N. Save and Vishakha Kelkar, " An Improved Image Compression Method using LBG with DCT", IJERT Journal, Volume-3,Issue-06, June-2014.

[7] Pallavi N. Save and Vishakha Kelkar, "An Improved Image Compression method using LBG with DCT and Fast search", International Conference on Advances in Computing and Information Technology, (ICACIT-2014) ISBN No: 978-93-5107-300-0, Elsevier Publication.

[8] Rafael C Gonzalez, Richard E. Woods, "Digital Image Processing"