Volume 5, Issue 2, February 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Image Compression using Efficient Codebook Generation

Srilakshmi Inuganti, Veerraju Gampala
Computer Science and Engineering, GMRIT
Andhra Pradesh, India

Abstract-- In recent years, the development and demand of multimedia products grow increasingly fast, which requires more bandwidth of networks and storage of memory devices. Therefore the need to minimize the amount of digital information stored and transmitted is an ever growing concern in the modern world. This can be achieved through the usage of digital image compression techniques. Vector Quantization (VQ) is a useful scheme for grayscale image compression.VQ can be divided into three parts: codebook generation, image encoding, and image decoding. In VQ, the reconstructed image quality is restricted by the codebook used in the image encoding/decoding procedures. In this paper variant of code book technique is used in order to improve the quality of image. Experimental results show our proposed technique outperforms previous methods.

***Keywords:** Image Compression, Vector Quantization, codebook, LBG algorithm.*

I. INTRODUCTION

Image Compression, an important area in the field of digital image processing, deals with techniques for reducing the storage required for saving an image or the bandwidth required for transmitting it [3]. The objective of Image Compression is to reduce irrelevance and redundancy of the image data thereby optimizing the storage space and increasing the transmission rate over WebPages. This paper focuses Vector Quantization which is a simple approach for image compression. VQ has been widely used in different applications such as Pattern Recognition,Image Compression,Speech Recognition, Face Recognition and soon .There are three major procedures in VQ, namely codebook generation, encoding procedure and decoding procedure.In Image encoding process each grayscale image to be compressed is first divided into a set of non-over-lapped image blocks of nxn pixels. The goal of codebook generation process is to generate a set of representative codewords to form a code book. LBG algorithm is most commonly used algorithm for codebook generation[4]. In the decoding procedure the compressed image of VQ can be reconstructed. In VQ the reconstructed image quality is limited by the codebook used.To provide better image quality in VQ large sized codebook is needed. To avoid using a large sized coodbook while achieving a better image quality expanded codebook is used[1-2]. Rest of the paper is constituted as follows: Section II focuses on Vector quantization and variant code book Section III focuses on encoding and decoding procedure. Results obtained and comparisons shown in Section IV. Finally, conclusion of the paper presented in Section V.

II. VECTOR QUANTIZATION

For the purpose of image compression, the operations of VQ include dividing an image into several vectors (or blocks) and each vector is mapped to the codewords of a codebook to find its reproduction vector.[5] In other words, the objective of VQ is the representation of vectors X ε Rk by a set of reference vectors CB = {C1;C2; : : : ;CN} in Rk in which Rk is the k-dimension Euclidean space.[7] CB is a codebook which has a set of reproduction codewords[6] and Cj = {c1; c2; : : : ; ck} is the j-th codeword. The total number of codewords in CB is N and the number of dimensions of each codeword is k. The following figure provides overview of vector quantization.

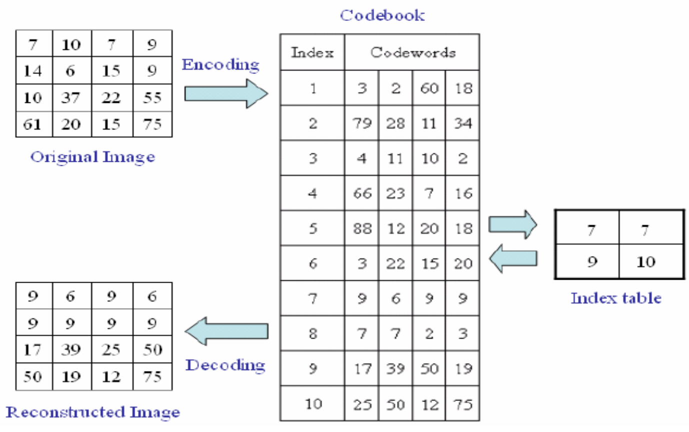


FIGURE 1. Overview of Vector Quantization

1. Proposed Codebook generation:

The following is the codebook variant algorithm and we have used mapping function of partition training vectors into n clusters.Let X=(x1,x2,x3,……xk) be a training vector and d(X,Y) be the Euclidean distance between any two vectors.[2]

Step 1: Randomly generate a initial codebook CB0.

Step 2: i=0.

Step 3: Perform the following process for each training vector.Compute the Euclidean distances between the training vector and the codewords in CBi. The Euclidean distance is defined as

$$d(X, C) = \sqrt{\sum_{t=1}^{k} (x_t - c_t)^2}$$

Step 4: Partition the codebook into N cells.

Step 5:Compute the centroid of each cell to obtain the new codebook CBi+1.

Step 6:Compute the average distortion for CBi+1. If it is changed by a small enough amount since the last iteration, the codebook may converge and the procedure stops. Otherwise, i = i + 1 and go to Step 3.

Suppose the VQ codebook CB = {cw0, cw1, …, cwN–1} of N codewords was previously generated by using the accelerated version of the LBG algorithm. Let F denote the expanding factor. To generate the expanded codebook ECB of N×F codewords, where ECB = {ecw0, ecw1, …,ecwNF–1}, the codewords in CB are first sorted by their sum values in ascending order.[8] Each codeword cwi of CB is copied and stored in ecwi_F of ECB. The remanding codewords in ECB are generated by linear prediction of these copied codewords. The following equation is used to generate j−th codeword ecwj in ECB where j is less than or equal to F×(N – 1)[9]

$$ecw_i = \begin{cases} cw_{\left\lfloor \frac{j}{F} \right\rfloor} & \text{if } j \bmod F = 0 \\ \dfrac{F - j \bmod F}{F} \times cw_{\left\lfloor \frac{j}{F} \right\rfloor} + \dfrac{j \bmod F}{F} \times cw_{\left(\left\lfloor \frac{j}{F} \right\rfloor + 1\right)} & \text{otherwise} \end{cases}$$

In this equation, N codewords in ECB are directly copied form CB in the first case. In addition, each remaining codeword in ECB is computed based on the linear prediction technique in the second case. Note that some codewords in ECB cannot be generated by using Eq. (2). In fact, a total of (F – 1) codewords in ECB cannot be processed in Eq. (2). Such expanded codewords can be generated by using the extrapolation technique. An example of codeword mapping from CB to ECB is depicted in Fig. 1

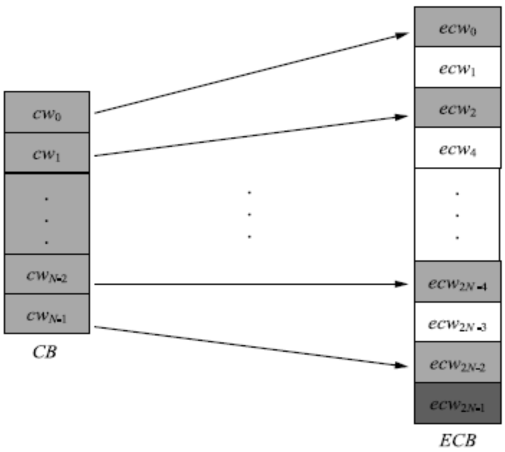


Figure 2. Example of codeword mapping from CB to ECB with F=2

The given image is partitioned into a set of non−overlapped image blocks of n * n pixels. Image blocks are sequentially processed in the order of left−to−right and then top−to−bottom.If the image block x to be processed is either in the first column or row of the image, it is then compressed by VQ using the expanded codebook. Let Ix denote the index of the closest codeword in the expanded codebook for x.[9] The index[4] Ix is further processed by the relatively addressing technique.

III. RESULTS AND DISCUSSIONS

Results

Four grayscale images, which are commonly used in the community of image processing and compression, were used to do experiment in this paper. The evolution of the proposed method is made on the basis of the results of compression, image quality and CPU processing time on the four images.

The compression ratio is defined as the ration of the number of bits in the original image file to the no.of bits in the compressed file. The quality of image is determined by the peak signal-to-noise ratio (PSNR) which determines the difference between the two images. It is defined as PSNR= 20log10 (b/rms) db, where b is the largest possible values of the signal or pixl value, and the rms is the root-mean-square difference between two images. The PSNR is given in units of decibel (dB), which measures the ration of the peak signal and difference between two images. That is to say, the higher the value, the better the image quality. For each image, the results are measured for all the four methods. It is observed that proposed method consistently converges faster than the other methods by maintaining compression ration and image quality.

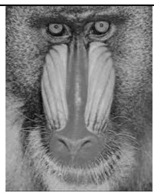
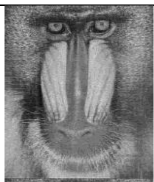
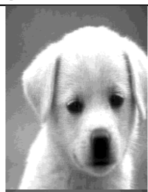
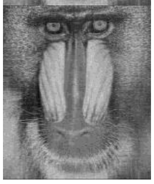
Original image				
Compressed images				
VQ				
Proposed VQ				
PSNR value				
VQ	81.4625	74.5902	80.6327	83.7712
Proposed VQ	81.5718	73.9736	80.1075	83.5732
Compression Ratio				
VQ	0.7187	0.9083	0.7826	0.2616
Proposed VQ	0.7062	0.841	0.7826	0.2591
Speed(sec)				
VQ	9.023311	1.689906	5.84489	1.244554
Proposed VQ	4.336461	0.779088	2.81434	0.560941

Fig.1 Original images, compressed images of multiplicative update NMF and proposed CNMF compressed images with their PSNR value, Compression ratio and convergence speed

IV. CONCLUSION

In this paper we reported image compression based on efficient code book. This method increases the compression ratio by 10% than VQ method. Furthermore, this converges fast. The compression results were experimentally shown. The proposed method can be integrated with other techniques to get better results.

REFERENCES

[1] Y.C. Hu and C.C. Chang, "Efficient greyscale image compression technique based on vector quantization", OPTO−ELECTRONICS REVIEW 19(1),versta 104–113, (2003).

[2] Tzu-Chuen Lu, Ching-Yun Chang," A Survey of VQ Codebook Generation" *Journal of Information Hiding and Multimedia Signal Processing,* Ubiquitous International 2010 Volume 1,

[3] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design", *IEEE T. Commun.* 28, 84–95 (1980).

[4] S.X. Chen, F.W. Li, and W.L. Zhu, "Fast searching algorithm for vector quantization based on features of vector and subvector", IET Image Process (2008).
[5] Y.C. Hu, B.H. Su, and C.C. Tsou, "Fast VQ codebook search algorithm for greyscale image coding", Image Vision Comput. (2008).
[6] C.C. Chang, G.M. Chen, and Y.C. Hu, "A novel index coding scheme for vector quantization", Fundam. Inf. (2006).
[7] Y.C. Hu, C.C. Lin, and K.L, Chi, "Block prediction vector quantization for greyscale image compression", Fundam.Inf.(2007).
[8] K. Sasazaki, S. Saga, J. Maeda, and Y. Suzuki, "Vector quantization of images with variable block size", Appl. SoftV Comput., (2008).
[9] Rafael C.Gonzales and Richard E.Woods, Digital image processing, Pearson Education, 2001, 2nd edition