AN EFFICIENT ALGORITHM FOR CODEBOOK DESIGN IN TRANSFORM VECTOR QUANTIZATION

Momotaz Begum, Nurun Nahar, Kaneez Fatimah, M. K. Hasan, and M. A. Rahman*
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Bangladesh
*Faculty of Engineering and Applied Science, Memorial University of Newfoundland, Canada
E-mail: khasan@eee.buet.ac.bd

ABSTRACT

In this paper, a new wavelet-domain codebook design algorithm is proposed for image coding. The method utilizes mean-squared error and variance based selection schemes for good clustering of data vectors in the training space. As the clustering process terminates only in two steps, it is highly computationally efficient as compared to other reported methods. Simulation results are presented to show the superior performance of the proposed method in terms of peak signal-to-noise ratio as compared to the standard Linde-Buzo-Gray algorithm for codebook design.

Keywords

Image coding, vector quantization, codebook design, feature-based selection, wavelet-domain.

1. INTRODUCTION

Vector quantization (VQ) [Cos96]–[Gra92] is an effective means for lossy compression of speech and digital images. The basic idea behind all VQ based image compression techniques are similar and in general an m–dimensional vector quantizer Q may be defined as Q: $R^m \rightarrow \gamma$ where R^m is the m-dimensional Euclidean space γ is a subset of R^m, $\gamma = \{c_1, c_2,\ldots,c_n\} \subset R^m$, and termed as an n-dimensional codebook. The output vectors $\{c_i \in R^m\}_{\{i=1, 2,\ldots,n\}}$ are referred to as codevectors The index numbers corresponding to them are transmitted through a channel. Cost of transmitting an image is decided by the value of n and the quality depends upon the goodness of the codebook. Both the encoder and decoder work using the same codebook. In VQ systems the encoder determines the closest codeword in the codebook against an incoming vector and thereby mapping the input vectors into a set of index numbers. The decoder's function is only a table look up-fetching out the codevectors corresponding to the received index numbers. This scheme is called full-search VQ. Several approaches have been reported for image coding by VQ of transform coefficients rather than the image pixels [Kar98]–[Aver96] and are referred to as transform vector quantization (TVQ). TVQ–based approaches combine the merits of VQ and transform coding. Generally, they utilize the standard Linde-Buzo-Gray (LBG) algorithm [Lin80] for codebook design.

WSCG POSTERS proceedings
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

In order to alleviate problems associated with the LBG algorithm, we propose a new method for codebook design in the wavelet-domain using partially the essence of genetic algorithm (GA) [Del95], [Pan95]. Here, a number of clusters are formed in the training vector space, centroids to which constitute the codebook. This is similar to the conventional LBG algorithm. But the sector where our approach differs from the LBG is the process of cluster formation. Clusters are generated on the basis of requirements of purpose rather than emerged as an implicit property of clustering algorithm. From any initial condition similar type of vectors in the training space are forced to form clusters and the cluster representatives, after going through some fitness evaluation, constitute the codebook. This fitness evaluation is a popular strategy in GA to determine the degree of goodness of an individual. To preserve the original feature of vectors, we avoid taking the mean value of clusters as the cluster representative until a good clustering is ensured. This increases the effectiveness of the codebook. Unlike the LBG algorithm, the proposed method is free from the severe initialization problem and requires less computational time to give a better solution.

2. PROPOSED ALGORITHM

Initially, a training set is created using the transform coefficients of image data. A set of training images are first classified into edge and non-edge groups using 2-D wavelet transformation. One level decomposition of images using 'Daubechies 3' as mother wavelet decomposes each image into four different subbands namely, LL (approximate), HL (horizontal), LH (vertical) and HH (diagonal) and they represent the original image at different level of resolution. The horizontal, vertical and diagonal images contain the horizontal, vertical and diagonal edge information, respectively, while the approximate image contains the low frequency contents of the original image. Each component image is divided into sub-image blocks of size $m \times m$ which are then converted to m^2–dimensional vectors. Collection of these vectors are considered as pattern-space and is called training vector set. There are four training sets for a single image and four separate codebooks are designed corresponding to each training set.

Evolution of the codebook

For a particular training set (e.g., training set corresponding to approximate component) a step by step selection process is applied to generate a prescribed number of temporary clusters. Unlike GA, this is a strictly deterministic selection with one selection criteria associated to each step. 'Similar' vectors of the training space gather under the same cluster. Mean−squared error (MSE) between vectors are taken as the similarity measure. Clustering process is initiated by finding the smoothest vector in the training space. The variance of a vector is used as the determining factor of smoothness and the vector having minimum variance is selected as the smoothest vector and termed as the reference vector. For any other vector, increase in MSE distance with the reference simply indicates the presence of more feature variation compared to the reference one. Training vectors are then classified into C different clusters according to their MSE distances with the reference. From these clusters, C vectors are chosen very carefully, one from each cluster, by natural selection–a well known genetic operator. Two selection criteria are specified at this stage: one is based on MSE measure and the other on variance measure defined as given below.

- *MSE based selection*: Any vector selected from a temporary cluster should be the best representative vector of its own population. Degree of goodness is measured by a cost function defined as

$$D(X_i^l, Y^l) = \frac{1}{m^2} \sum_{k=1}^{n_c} \sum_{j=1}^{m^2} \left(x_{ij} - y_{kj}\right)^2, \quad i = 1, \cdots, n_c \quad (1)$$

where X_i^l is the candidate vector and Y^l is a member of the l-th cluster, m^2 is the total number of elements in a vector, n_c is the total number of vectors in the l-th cluster. The vector X_i^l is selected as the representative vector of the l-th cluster for the value of i for which D is minimum. Using this selection process C_1 vectors are selected from $C = C_1$ clusters, one from each cluster. MSE−based selection ensures that bad clustering (to some degree) at this stage will not eventually affect the cluster representative selection. If due to bad clustering, a few members of high variance vectors get included within a cluster of low variance vectors then MSE-based selection will ultimately select a vector from the low variance region as the cluster representative due to their dominancy in the cluster. Also as we do not take the mean value of vectors as cluster representative, the original features of the vectors are preserved.

- *Variance based selection*: Vector having the maximum feature variation is chosen as the representative of a cluster. Here, variance is chosen as the determining factor for feature estimation and the vector having maximum variance is selected. Using this selection process, C_2 vectors are selected from C_2 clusters, one from each cluster.

Thus, finally we get $C_{total} = C_1+C_2$ representative vectors which are subjected to a test so that degeneracy (case with more than one codevectors representing the same solution) can be avoided. Any vector, once selected, must check itself against all other vectors selected previously. If its MSE distance with any one of them fall below a threshold value ε, new selection is discarded. Thus for the best representative of a temporary cluster to be included in the C_{total} selected vectors queue must satisfy $\|X^l - Z^m\| > \varepsilon$ (m=1,2,....,(l–1)), where X^l is the representative vector of the l-th cluster undergoing selection test and Z^m denotes the already selected representative from the previous (l–1) clusters. Let, there exist C_U unique vectors after this test, where $C_U \leq C_{total}$. C_U real clusters are then formed. Initially, each of the C_U vectors is the only member of the corresponding cluster. Next, the vectors of training space are allowed to enter in any one of the C_U clusters such that their MSE with the centroid of a cluster is minimized. The centriod is defined as the mean of a cluster. This eliminates any error, if exists, during the selection of a cluster representative at the end of temporary cluster formation. Vectors that had to select an unmatched vector as the group representative due to bad clustering, now would have a chance to choose a better match. This contributes in making a good clustering. Each time a new vector is included into a cluster, its centroid is readjusted. Thus finally we have C_U real clusters and C_U centroids. The codevectors corresponding to C_U centroids are the best among all other codevectors in the training space. Next, a fitness evaluation is performed on them.

Fitness test

To each one of the C_U vectors we associate a value termed as the 'fitness score'. For a vector, a fitness score m simply indicates that it has been selected as 'closely-matched' (among the C_U vectors) for m times by a large set of vectors. This new vector-set is called fitness vector and is created in the same way as training-vector space but from different images. Use of a set of images for fitness evaluation is unlike other GA-based algorithm for codebook design. A fitness vector will select a candidate vector as 'closely–matched' whenever its MSE with this candidate is minimum compared to that with all other candidates in the C_U vectors queue. The vectors having fitness score greater than a predefined threshold, e.g. α_1, are selected as the member of the final codebook. The threshold α_1 is a tuning parameter to vary the bit rate. Value of it is decided on a trial and error basis to make a compromise between PSNR and the length of codebook which ultimately determines the bit rate.

3. EXPERIMENTAL RESULTS

In this section, we present computer simulation results to evaluate the performance of the proposed codebook design algorithm. Here, codebook is designed using ten standard grayscale images of size 256×256 as training images and six other images for fitness evaluation. Both training and fitness vectors are of dimension 1×25. Two standard images `Lena' and 'Pepper' which were neither in the training set nor in the fitness set are used as test images. Values of different parameters (e.g., α_1, C_1, C_2 etc.) were varied until the best performance is obtained. Performance is quantified by peak signal-to-noise ratio (PSNR) and bit rate which are defined as

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

$$\mathrm{MSE} = \frac{1}{K \times K}\sum_{i=1}^{P}\sum_{j=1}^{P}\left(X_{i,j} - Y_{i,j}\right)^2 \quad (3)$$

where $K \times K$ is the total number of pixels in the image and $X_{i,j}$ and $Y_{i,j}$ represent (i,j) pixel values in the original and reproduced images, respectively. Each coding index is represented by the least integer number of bits to calculate the entropy coded bit rate and is referred to as BR_2. Bit rate without entropy coding is referred to as BR_1 and depends upon the block size and the required number of bits to encode a single block. For a block size of $m \times m$, if the number of bits required to encode that block is b then BR_1 is b/m^2 bits/pixel

Lena		**Pepper**	
PSNR (dB)	**Bit Rate (bpp)**	**PSNR (dB)**	**Bit Rate (bpp)**
26.2256	0.0530	27.2090	0.0875
27.5580	0.0861	28.6397	0.1233
28.8281	0.1213	29.5757	0.1581
29.6210	0.1550	30.2452	0.1963
30.2556	0.1913	31.2046	0.2651
31.3982	0.2550	32.2232	0.3713
32.2665	0.3091	32.9466	0.4798

Table 1. Results for test images Lena and Pepper

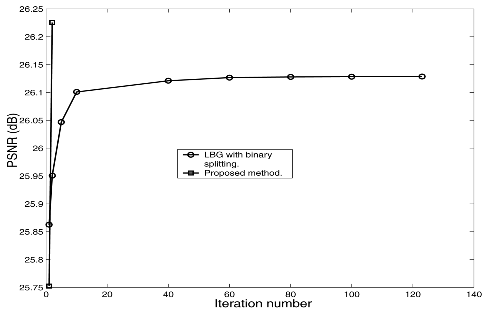

Figure 1. Convergence characteristics of the LBG and proposed algorithms.

Figure 2. Coding results for test image 'Lena': (a) Original; (b) 0.1892 bits/pixel, 29.3270 dB; (c) 0.2666 bits/pixel, 31.4672 dB; (d) 0.4540 bits/pixel, 33.4164 dB.

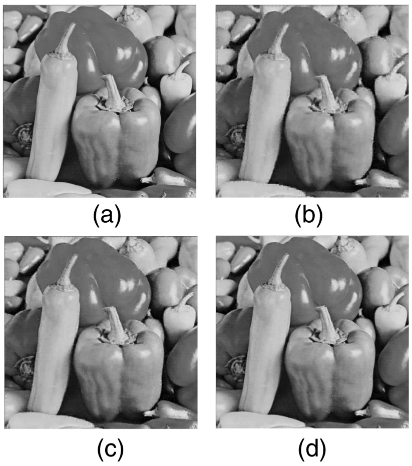

Figure 3. Coding results for test image ‘Pepper': (a) Original; (b) 0.1213 bits/pixel, 28.8281 dB; (c) 0.1913 bits/pixel, 30.2556 dB; (d) 0.3091 bits/pixel, 32.2665 dB.

Table 1 presents the performance of the proposed method in terms of PSNR and bit rate. As can be seen bit rate vs. PSNR performance of the proposed method is very good. Figure 1 shows the convergence characteristics of our proposed method with the standard LBG algorithm initialized by famous binary splitting technique. Starting with a PSNR value of 25.8626 dB, LBG loops over 123 iterations and takes approximately 25 hours to finally reach at a PSNR value of 26.1285 dB. Size of the codebook is 256 and the corresponding bit rate (BR_1) is 0.52 bits/pixels. But the proposed method gives a PSNR of 26.2256 dB at the same bit rate but within just two steps taking only 1.8 hours. At the first step, the representatives of temporary clusters form the codebook and give a PSNR of 25.89 dB at a bit rate (BR_1) of 0.5897 bits/pixel. At the next (and final) step the representatives of real clusters constitute the codebook and give a better performance than the LBG. In both cases, simulation was carried out on a Pentium IV PC with processor speed of 1.5 GHz. When the clusters generated by the proposed algorithm are used to initialize the LBG algorithm, LBG after 18 iterations exhibits only a 0.0136 dB improvement in PSNR value while taking 4.2 hours to converge. PSNR value according to our proposed method is 27.558 dB at a bit rate (BR_1) of 0.52 bit/pixel while after initializing the LBG with the generated clusters, PSNR value becomes 27.5716 dB at the same bit rate.

Figures 2 and 3 show the reconstructed images ‘Lena' and ‘Pepper', respectively, at different bit rates using the proposed method. The original images are also included for subjective quality measure. As shown, the quality of the reproduced images are satisfactory.

4. CONCLUSION

An effective method for codebook design in the wavelet-domain has been proposed. The codevectors are chosen as centroids of a set of clusters formed using the MSE and variance based selection strategy. The two major drawbacks of the LBG algorithm namely, the choice of initial codebook and the huge computational burden, have been alleviated. Results show superior performance of our method as compared to that of the standard LBG algorithm.

5. REFERENCES

[Cos96] Cosman, P. C., Gray, R. M., and Vetterli, M. Vector quantization of image subbands: a survey. Trans. IEEE Image Proc., Vol. 5, No. 2, pp. 202-225, 1996.

[Lin80] Linde, Y., Buzo, A., and Gray, R. M. An Algorithm for Vector Quantizer Design. Trans. IEEE Commun., Vol. COM-28, pp. 84-95, 1980.

[Gra92] Gray, R. M., and Gersho, A. Vector quantization and signal compression. Norwell, MA: Kluwer, 1992.

[Kar98] Karayiannis, N. B., Pai, P. I., and Zervos, N. Image compression based on fuzzy algorithms for learning vector quantization and wavelet image decomposition. Trans. IEEE Image Proc., Vol. 7, No. 8, pp. 1223-1230, 1998.

[Ant92] Antonini, M, Barlaud, M., Mathieu, P., and Daubechies, I. Image coding using wavelet transform. IEEE Trans. Image processing, Vol. 1, pp. 205-220, 1992.

[Aver96] Averbuch, A., Lazar, D., and Israeli, M. Image compression using wavelet transform and multireslution decomposition. Trans. IEEE Image Proc., vol. 5, pp. 4-15, 1996.

[Del95] Delport, V., and Koschorreck, M. Genetic algorithm for codebook design in vector quatization. Electron. Lett., Vol. 31, No. 2, pp.84-85, 1995.

[Pan95] Pan, J. S., McInnes, F. R., and Jack, M. A. VQ codebook design using genetic algorithm. Electron. Lett., Vol. 31, No. 17, pp.18-19, 1995.