FVQEOPT: Fast Vector Quantization Encoding with Orthogonal Polynomials Transform

Krishnamoorthy R. and Punidha R.

***Abstract*—This paper deals with a design of new vector quantization algorithm for coding of color images in the Transform domain. In order to speed up the design issue, the feature of transform coding is combined with vector quantization. The transformed training set is obtained with the proposed integer Orthogonal Polynomials transform with reduced computational complexity. The proposed method then generates a single codebook for all the three color components, utilizing the inter-correlation property of the individual color plane as well as interactions among the color planes with the proposed transformation. In the codebook generation phase of the proposed vector quantization encoding, binary tree method is used to achieve considerable saving in codebook construction time. The vector encoding phase uses binary search and partial distortion elimination to further reduce the encoding time for finding closest codeword of an input vector. The experimental result shows that the proposed algorithm greatly reduces the encoding time. The proposed algorithm is also compared with existing standard LBG algorithm.**

***Index Terms*—Vector quantization, orthogonal polynomials transform, binary tree method.**

I. Introduction

In recent years, Vector Quantization (VQ) has been found to be an efficient data compression technique for images [1]. In the literature, the different types of VQ techniques such as Classified VQ [2], Address VQ [3], Adaptive VQ [4] have been reported for various purposes. Vector quantization based image compression [5-7] exhibits good compression performance at low bit rates while computational complexity has been always a tough problem for its time consuming encoding system. Vector quantization in frequency domain also called as Transformed Vector Quantization (TVQ) is reported by combining both transform coding and VQ that takes advantage over VQ in spatial domain. The transformed vector components in high frequency regions are low energy components and can be discarded. Consequently the dimension of transformed vectors and the complexity of VQ are both reduced. This yields a reduced codebook size and hence a higher compression ratio than that of a VQ alone. Traditionally, VQ encoding contains two phases: codebook generation and closest codeword search. In the codebook generation phase, the most popular method used is the Linde–Buzo-Gray (LBG) algorithm [8], which is also called the Generalized Lloyd Algorithm (GLA).

Techniques for fast and efficient codebook generation of vector quantization have been reported in the literature. In LBG based codebook construction, various improvements [9, 10] have been adopted to minimize the computing time and to generate better codebook for representing the input vector. A code book generation algorithm using Pairwise Nearest Neighbour (PNN) has been proposed by Equitz [11], which begins with separate cluster for each vector in the training vectors and merges together two clusters at a time until desired number of codebook size is achieved. An adaptive vector quantization for coding of color images is proposed in [12]. But its disadvantage is that, it needs color conversion matrix and interactions among the color planes are not taken into account. In [13], wavelet based color image compression is implemented using tree structured vector quantization by Annadurai et. al. Wherein, the RGB color space is converted into YUV color model and quantization is applied as if there are three monochrome images and interactions among color planes are not taken care.

The second stage of VQ encoding is the closest codeword search algorithm, where the index of codeword with minimum distortion is obtained from the codebook and is entropy coded for further compression. In order to find the best-matched codeword in the encoder, the ordinary VQ coding scheme employs the full search algorithm, which examines the Euclidean distance between the input vector and all the code words in the codebook. Hence the encoding time complexity in the full search algorithm is given as $O(kn)$ where k is the input vector dimension and n is the codebook size, and this complexity grows as the input vector dimension increases. To overcome this problem, few fast codeword search algorithms have been reported in [14, 15]. For fast code book search method, the performance of mean-distance-ordered partial codebook search (MPS) algorithm is improved in [16] by combining MPS and triangular inequality elimination (TIE) to accelerate the closest codeword search process. To summarize, both the VQ encoding phases such as codebook generation and codeword search algorithm should be optimized in terms of computation time. Performance of vector quantization in transformed domain results in lesser computational complexity and reduced memory requirement, when combined with binary tree codebook. Hence in this work, a binary tree codebook is constructed with the proposed orthogonal polynomials transformed training vectors. Since, the proposed transform is configured to take not only the interactions in the individual color planes but also takes into account interactions between the color planes, there is no need to design codebook for three color components separately unlike three separate code books designed in the existing literatures. Also color coordinate conversion

Manuscript received August 4, 2012; revised October 25, 2012.
The authors are with Computer vision lab, Department of CSE, Anna University of Technology, Tiruchirappalli - 620 024, Tamil Nadu, India (e-mail: arkrish26@hotmail.com, br_punidha@yahoo.co.in).

DOI: 10.7763/IJCTE.2013.V5.641

transformation is not required in the proposed scheme. The transformed binary tree codebook has the advantage that the computational complexity grows linearly with rate, rather than exponentially as in GLA. Once the codebook is constructed using Binary Tree Method (BTM) algorithm, the second phase of VQ encoding involves closest codeword search. The search algorithm in this proposed work uses binary search and Partial Distortion Elimination (PDE) for finding closest code vector of an input vector.

This paper is organized as follows: The proposed Orthogonal Polynomials based transform coding of color images is presented in section 2. The basis operator of the proposed transform coding for reconstruction purpose is given in section 3. The proposed fast vector quantization encoding using orthogonal polynomials transform is given in section 4. The experimental results and comparison with existing techniques are presented in section 5. Conclusion is presented in section 6.

II. Orthogonal Polynomials Based Transform Coding

In order to devise a color image transform coding technique, we first analyze the image formation system. As per the classical definition, the color image formation can be described as

$$I(x,y,z)=\iiint f(\xi,\eta,\gamma)\,d\mu(\xi)d\mu(\eta)d\mu(\gamma) \tag{1}$$

where the object function $f(\xi,\eta,\gamma)$ is integrable on a measure space and μ is a σ finite measure with an infinite number of points of increase. The image I can be considered to be a signed measure on the ring of all measurable sets. It can be easily shown that if I is non-negative and finite valued on the σ ring $\Re$ of measurable sets then f can be defined as a kind of derivative of I relative to a signed measure supported by a null set. Expressing object function f in terms of derivatives of the image function I relative to its spatial and color coordinates is very useful in connection with devising a color image transform coding technique. Since representation of f in terms of derivatives of I is considered to be an ill-posed problem [17], it is desirable that differentiation of I must undergo a smoothing process. The underlying smoothing operation can be either of the following two types of linear transformation:

1) The integral convolution transformation

$$f(x)=\int\delta(x-t)\,I(t)dt \tag{2}$$

2) The ordinary linear transformation

$$f_i=\sum_{k=0}^{n-1}u_{ik}I_k,(i=0,\ ...,\ n-1) \tag{3}$$

The linear transformation defined in Eq. 2 by the matrix $|\delta(x_i-t_i|)$ is a smoothing operation provided it is totally positive, whereas the linear transformation defined in Eq. 3 by the matrix $U=|u_{ik}|$ is a smoothing operation provided it is totally positive. The matrix U is totally positive provided all the minors of all orders of its determinant $|u_{ik}|$ are non-negative. The point-spread function $M(s, x)$ is considered to be a real valued function defined for $(s, x) \in S \times X$ where S and X are ordered subsets of real values. In the case where S consists of a finite set $\{0, 1, ..., n-1\}$, the function $M(s, x)$ reduces to a sequence of functions

$$M(i,x)=u_i(x),\quad i=0,1,...,n-1$$

Consequently the linear three dimensional image transformations can be shown as

$$\beta'(s,\zeta,\eta)=\int_{x\in X}\int_{y\in Y}\int_{z\in Z}M(\zeta,x)M(\eta,y)M(s,z)I(x,y,z)dxdydz \tag{4}$$

The point-spread function $M(s,t)(M(i,t)=u_i(t))$ is a row, column and color smoothing operation provided the set of functions $\{u_0(t),...,u_{n-1}(t)\}$ is a T-system over a closed interval [a, b]. Considering each of S, X, Y and Z to be finite set of values {0, 1, …, n-1}, the matrix notation of Eq. 4 is

$$\left|\beta'_{ijk}\right|_{i,j,k=0}^{n-1}=(|M|\otimes|M|\otimes|M|)^t|I| \tag{5}$$

where the point spread operator $|M|$ is

$$|M|=\begin{vmatrix} u_0(x_0) & u_1(x_0) & \cdots & u_{n-1}(x_0)\\ u_0(x_1) & u_1(x_1) & \cdots & u_{n-1}(x_1)\\ & \vdots & & \\ u_0(x_{n-1}) & u_1(x_{n-1}) & \cdots & u_{n-1}(x_{n-1})\end{vmatrix} \tag{6}$$

$\otimes$ is the outer product and $|\beta'_{ijk}|$ be the n^3 matrices arranged in the dictionary sequence that takes the effect of individual R,G and B planes as well as interactions among the color planes. $|I|$ is the image and $|\beta'_{ijk}|$ be the coefficients of transformation.

We consider the set of orthogonal polynomials $u_0(x)$, $u_1(x)$, ..., $u_{n-1}(x)$ of degrees 0, 1, 2, ..., $n-1$, respectively. The generating formula for the polynomials is as follows.
where

$$b_i(n)=\frac{\langle u_i,u_i\rangle}{\langle u_{i-1},u_{i-1}\rangle}=\frac{\sum_{x=1}^{n}u_i^2(x)}{\sum_{x=1}^{n}u_{i-1}^2(x)} \tag{7}$$

and

$$\mu=\frac{1}{n}\sum_{x=1}^{n}x \tag{8}$$

Considering the range of values of x to be $x_i = i$, i = 1, 2, 3, …, n, we get

$$b_i(n) = \frac{i^2(n^2 - i^2)}{4(4i^2 - 1)}, \quad \mu = \frac{1}{n}\sum_{x=1}^{n} x = \frac{n+1}{2} \quad (9)$$

The point-spread operators $|M|$ of different size can be constructed from Eq. 6 for $n \geq 2$ and $x_i = i$. For the convenience of point-spread operations, the elements of $|M|$ are scaled to make them integers and hence the proposed coding involves only integer arithmetic.

III. The Orthogonal Polynomials Basis

In case of R-G-B color space, the elements of the finite set Z for convenience can be labeled as {1, 2, 3}. For the sake of computational simplicity, the finite Cartesian coordinate set S, X and Y are also labeled in the identical manner. The point spread operator in Eq. 5 that defines the linear orthogonal transformation of color images can be obtained as $|M| \otimes |M| \otimes |M|$ in which $|M|$ can be computed and scaled from Eq. 7 as follows.

$$|M| = \begin{vmatrix} u_0(x_0) & u_1(x_0) & u_2(x_0) \\ u_0(x_1) & u_1(x_1) & u_2(x_1) \\ u_0(x_2) & u_1(x_2) & u_2(x_2) \end{vmatrix} = \begin{vmatrix} 1 & -1 & 1 \\ 1 & 0 & -2 \\ 1 & 1 & 1 \end{vmatrix} \quad (10)$$

The set of 27 three dimensional polynomial basis operators O_{ijk} $(0 \leq i, j, k \leq n\text{-}1)$ can be computed as

$$O_{ijk} = \hat{u}_i \otimes \hat{u}_j \otimes \hat{u}_k \quad (11)$$

where $\hat{u}_i$ is the $(i+1)^{st}$ column vector of $|M|$. The operator O_{ijk} is arranged in the dictionary sequence in such a manner that it becomes the $(i \times 3^2 + j \times 3 + k) + 1^{st}$ column vector of the point-spread operator $|M| \otimes |M| \otimes |M|$ in Eq. 5.

IV. Proposed Codebook Generation and Codeword Matching

The proposed VQ encoding algorithm called Fast Vector Quantization Encoding with Orthogonal Polynomials Transform (*FVQEOPT*) is presented in this section. This proposed algorithm consists of two steps namely extraction of features with orthogonal polynomials transform and design of BTM for codebook generation. It is observed that in the transformed domain, the high energy components are concentrated in the low frequency region. This means that the transformed vector components in the high frequency regions have very little information and hence these low energy components might be discarded entirely.

The BTM codebook generation algorithm exhibits higher computational efficiency and lower memory space, by focusing on the tasks of choosing the effective feature as split key and a good decision principle at each node to obtain better balanced tree structure. For binary tree partitioning, a single feature with largest variance is used as split key which is extracted from the training vectors. The rationale behind this choice is that if the data are very spread out along a particular feature, then presumably differences in that feature are more significant than differences in another more densely grouped feature. Once the split key has been chosen, the decision principle, to be decided upon, is a simple threshold operation. The mean value that corresponds to the split key feature is used as the threshold (*TH*) because the mean not only contains most of the statistical information of the feature but also requires less computation. The threshold *TH* partitions the training feature vectors at each non-terminal node into two halves. If the key value of a vector is less than the split threshold, then vector is placed on the left child node; otherwise, it is on the right child node. The splitting procedure is repeated until the desired number of clusters i.e. the desired codebook size is reached. The number of clusters is equal to the number of leaves at the lowest level of the binary tree. Finally, the code words of the codebook are formed by computing the centroids of the feature vectors falling at each node.

In the second phase of VQ encoding, binary searching and PDE is implemented for fast code word search. For binary searching, all the code words in the codebook are sorted based on the first component of code word. The lower and upper limit of binary searching is set as first and last code vector of the code book. For each input vector, choose the codeword say m with minimum distortion based on the first component of input vector as well as code word from the codebook (partial distortion). This partial distortion is compared with next code vector ($m+1$), and if this distortion is greater, then the upper limit of search is set ($m+1$) for binary search. This process is continued with second, third and so on with component of input vector and code vector until the upper and lower bounds exceed each other. Finally the closest codeword c_i is found with index *Ind*, which is further compressed using entropy coding, to result in compressed data stream. The computational complexity of proposed algorithm is greatly reduced as compared to FS technique. In general, the computation complexity of full search VQ encoding for color image is O(ckn), where k is the dimension of the vector, n is the codebook size and c is three color components. But the proposed algorithm reduces the computational complexity of codebook matching from O(ckn) to O($dlogn$), where d is the reduced dimension of transformed coefficients β'_{ijk}. The value of c is also reduced to 1, since single codebook is generated for all the three color components utilizing the inter-correlation property of the color planes due to the proposed transform. The proposed *FVQEOPT* technique is presented as an algorithm hereunder.

The proposed FVQEOPT Algorithm:

Input: Color image having three components R, G, and B, each of size ROW$\times$ COL. [ ] denotes the matrix and the suffix denotes the elements of the matrix. Let $|M| = |M| \otimes |M| \otimes |M|$ be the 3-D polynomial operator of size (27×27), and [I] be the (3×3) color image region extracted from each of the R, G, B components, arranged in dictionary sequence, so as to take into account interactions among color planes.

Output : Encoded color image with *FVQEOPT* algorithm.

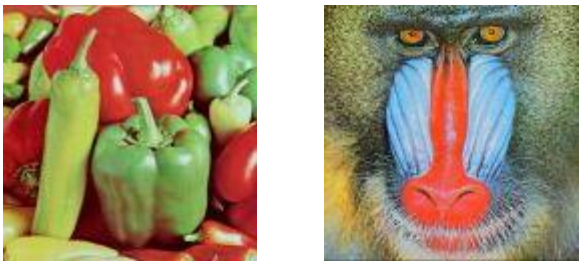
Steps:

1) Partition the input image into non-overlapping blocks of size ($m \times m$) (typically m = 3) and obtain input block of pixel values I_j $\{ j = 1, 2 \ldots k;\ k = m \times m \times 3 \}$ from three individual R, G, B color components.
2) Obtain the total number of input training vectors V, with each vector representing the input block I from the image of size (ROW × COL) { where V=(ROW × COL ×3) / k }.
3) Compute the Orthogonal Polynomials transformed co - efficients $[\beta'] = [\mathcal{M}]^t [I]$ for all the training vectors V as described in section 2.
4) Discard the high frequency coefficients based on energy preserving property of the proposed transform coding to form d-dimensional the transformed training vectors X, where $X = \{x_{ij}, i = 1, 2, \ldots, V$ and $j = 1, 2, \ldots, d; d < k\}$ as described in section 4.
5) Construct the codebook using BTM algorithm with N code words and each codeword consisting of d-dimensional code vector represented as c_{ij} (i=1 to N, j=1, 2, ..., d, where $d < k$).
6) Arrange the codebook in ascending based on the first component of code words.
7) Use binary search and partial distortion elimination to find nearest code word of input transformed training vectors X.
8) Encode the closest code word index using entropy encoding for further compression.

END.

V. Experiments and Results

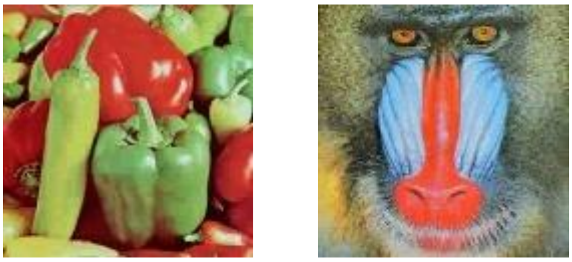
The proposed FVQEOPT algorithm for color image coding has been implemented in Intel Core(2)-Quad, 2.3 GHZ speed processor system with 2000 test images. Two sample standard test images viz. pepper and baboon images which are of size (256×256) with pixel values in the range of 0-255 in each of R, G and B color planes are shown in fig. 1(a) and 1(b) respectively.



(a) (b)

Fig. 1. Original test images

The input color images are partitioned into non-overlapping sub-images of size (3×3) in the three R, G, B color space, and the proposed transform coding is applied to obtain the transformed co-efficients β'_{ijk} as described in section 2. After discarding half of the high frequency transformed coefficients, the vector of d-dimensional transformed training vectors V is chosen for codebook training. The codebook is constructed with BTM method and obtain the best matched codeword and its index *Ind* from the codebook using binary search and partial distortion elimination as described in section 4. This index *Ind* is further encoded using entropy coding and obtain compressed bit streams. The decompression is done by first entropy decoding followed by inverse vector quantization, as a simple table look up process in the codebook. Finally, the inverse transform is applied on the de-quantized coefficients using orthogonal polynomials basis as described in section 3 and the reconstructed color image is obtained. The computation time of VQ encoding is calculated for both codebook construction and code words matching with the proposed algorithm for the sample test images. For VQ encoding time when the codebook size is 1024, the proposed algorithm took 0.143 seconds(s) and 0.137s for pepper and baboon images respectively. Similarly, for VQ encoding of codebook size 512, the proposed algorithm took 0.086s for pepper image and 0.092s for baboon image. In the same way, for VQ encoding when the codebook size is 256, the FVQEOPT algorithm takes 0.057s and 0.056s for the same standard test images. The performance of proposed vector quantization is measured with Peak-Signal-to-Noise-Ratio (PSNR). For the codebook of size 1024, the proposed algorithm achieves a PSNR of 32.47dB and 32.05dB for pepper and baboon images respectively. The reconstructed images corresponding to the images in fig. 1 with the proposed algorithm are presented in fig. 2(a) and 2(b). While the codebook size is 512, the PSNR of 30.67dB and 30.12dB are obtained with the proposed algorithm for pepper and baboon images. These results are presented in fig. 3(a) and 3(b) respectively.



(a) (b)

Fig. 2. Results of proposed vector quantization when the code book size is 1024.

(a) (b)

Fig. 3. Results of proposed vector quantization when the code book size is 512.

In order to measure the performance of the proposed vector quantization algorithm, we conduct experiments with LBG codebook generation algorithm. Here also the code word matching is implemented with binary search and partial distortion elimination. The computation time of Vector Quantization Encoding with the standard LBG (VQELBG) includes both codebook construction and code word matching. For VQ encoding for the codebook of size 1024, the VQELBG algorithm took 3.469s and 3.547s for pepper and baboon images respectively. Similarly for codebook of size 512, the VQELBG algorithm takes 1.953s for pepper image and 2.062s for baboon image. When the codebook size is 256, the VQELBG technique took 1.157s and 1.218s for pepper and baboon images. From the experimental results, it is evident that the proposed FVQEOPT encoding scheme takes less time, when compared with VQELBG.

VI. Conclusion

In this paper a new fast VQ encoding algorithm for color image transform coding is presented that uses Orthogonal Polynomials based transformation to extract the features of the training image. Due to the inter-correlation property of the proposed transform, a single codebook is generated for all the three color components of training image. The first phase of the proposed FVQEOPT encoding uses Binary Tree Method (BTM) to construct codebook, which reduces the computation time to a greater extent when compared with LBG codebook generation algorithm. The vector encoding uses binary search and partial distortion elimination to find the closest codeword search, which also reduces the computation time significantly. As a result, the proposed algorithm greatly reduces the computation time of VQ encoding, since it reduces both codebook generation time as well as closest codeword search time.

References

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1994.

[2] B. Ramamoorthy and A. Gersho, "Classified Vector Quantization of Images," *IEEE Transactions on Communications*, vol. 34, no. 11, pp. 1105-1115, 1986.

[3] M. N. Nasrabad and F. Yushu, "Image Compression using Address Vector Quantization," *IEEE Transactions on Communications*, vol. 38, no. 12, pp. 2166-2173, 1990.

[4] M. Goldberg, "Image Compression using Adaptive Vector Quantization," *IEEE Transactions on Communications*, vol. 34, no. 2, pp. 180-187, 1986.

[5] H. Al-Otum, W. Shahab, and M. Smadi, "Color Image Compression using a Modified Angular Vector Quantization," *Journal of Electrical Engineering*, vol. 57, no. 4, pp. 226- 234, 2006.

[6] W. Feng and S.Xiaoyan, "Image compression by Visual Pattern Vector Quantization," in *Proc. Data Compression Conference*, pp. 123-131, 2008.

[7] H. Chaur-Heh., S. Wei-Yang, and J. Ming-Haw, "Image Compression based on Multi-Stage Vector Quantization," *Journal of Visual Communication and Image Representation*, vol. 11, no. 4, pp. 374-384, 2000.

[8] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84-94, 1980.

[9] T. Chun-Wei and L. Chao-Yang, "A Fast VQ Codebook Generation Algorithm via Pattern Reduction," *Pattern Recognition Letters*, vol. 30, no. 7, pp. 653-660, 2009.

[10] L. Gang, L. Jing, and W. Quan, "A Robust Lin-Buzo-Gray Algorithm in Data Vector Quantization," *IEEE International forum on Information Technology and Applications*, vol. 1, pp. 464-467, 2009.

[11] W. Equitz, "A New Vector Quantization Clustering Algorithm," *IEEE Transactions on Acoustic, Speech, Signal processing*, vol. 37, no. 10, pp. 1568-1575, 1989.

[12] P. Boucher and M. Goldberg, "Color Image Compression by Adaptive Vector Quantization," *IEEE International Conference on Accoustic, Speech and Signal Processing*, vol. 9, Part. 1, pp. 539-542, 1984.

[13] S. Annadurai and M. Sunderasen, "Wavelet based color Image Compression relying on Subband Vector Quantization," *ICGST-GVIP Journal*, vol. 9, no. 1, 2009.

[14] C. H. Hsieh and Y. J. Liu, "Fast Search Algorithms for Vector Quantization of Images using Multiple Triangle Inequalities and Wavelet Transform," *IEEE Transactions on Image Processing*, vol.9, No. 3, pp.321-328, 2000.

[15] L. Torres and J. Huguet, "An Improvement on Codebook Search for Vector Quantization," *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 208-210, 1994.

[16] H. Yu-Chen, S. Bing-Hwang, and T. Chin-Chiang, "Fast VQ Codebook Search Algorithm for grayscale Image Coding," *Image and Vision computing*, vol. 26, no. 5, pp. 657-666, 2008.

[17] T. and A. N .Arsenin, *Solutions of Ill-posed Problems*, John Wiley and Sons, New York, 1977.

Dr. R. Krishnamoorthy is with computer Science and Engineering, Anna University of Technology – Tiruchirappalli. He obtained his Ph.D in Image Processing from the Indian Institute of Technology (IIT), Karaghpur in 1995 and M.Tech in Computer Science and Engineering from the Indian Institute of Technology (IIT), Kanpur, in 1992. His areas of Image Compression, Image Retrieval, Image encryption and authentication, Biometrics, Software engineering, Digital Watermarking, Steganography.

R. Punidha is a research student at Anna University of Technology - Tiruchirappalli, Tamil Nadu. She has received her M.Tech in Software Engineering from Periyar Maniammai Engineering College under Anna University Chennai in 2006 and got University First Rank. Her research interest includes Image & Video Compression, Image and Video Segmentation and Retrieval.