International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:11, 2009

# Codebook Generation for Vector Quantization on Orthogonal Polynomials based Transform Coding

R. Krishnamoorthi, N. Kannan

***Abstract*****—**In this paper, a new algorithm for generating codebook is proposed for vector quantization (VQ) in image coding. The significant features of the training image vectors are extracted by using the proposed Orthogonal Polynomials based transformation. We propose to generate the codebook by partitioning these feature vectors into a binary tree. Each feature vector at a non-terminal node of the binary tree is directed to one of the two descendants by comparing a single feature associated with that node to a threshold. The binary tree codebook is used for encoding and decoding the feature vectors. In the decoding process the feature vectors are subjected to inverse transformation with the help of basis functions of the proposed Orthogonal Polynomials based transformation to get back the approximated input image training vectors. The results of the proposed coding are compared with the VQ using Discrete Cosine Transform (DCT) and Pairwise Nearest Neighbor (PNN) algorithm. The new algorithm results in a considerable reduction in computation time and provides better reconstructed picture quality.

***Keywords*****—**Orthogonal Polynomials, Image Coding, Vector Quantization, TSVQ, Binary Tree Classifier

## I. Introduction

IMAGE compression addresses the problem of reducing the amount of data required to represent the digital image. One approach to image data compression is the use of unitary transformation that operates on the image to produce the set of transform coefficients. The simple and powerful class of transform coding is linear block transform coding, where the entire image is partitioned into a number of non-overlapping blocks and then the transformation is applied to yield transform coefficients. This is necessitated because of the fact that the original pixel values of the input image are highly correlated [1]. The international compression standard JPEG baseline system [2] uses the Discrete Cosine Transformation [3, 4]. In the early works, utilization of other transforms such as Haar [5], Slant [6], KL Transformation [7, 8], Hadamard [9] and Walsh [10] are also reported.

In the current scenario, the Vector Quantization has been found to be an efficient data compression technique for speech and image. In this, the image data is divided into non-overlapping blocks of same size (training vectors). The key to Vector Quantization is to construct a good codebook of representative vectors. The most popular method for designing a codebook was proposed by Linde, Buzo and Gray in [11, 12]. This method is now commonly referred to as LBG algorithm. In this algorithm, all the training vectors are clustered using minimum distortion principle around trial code vectors. The centroids of these clusters then become the new trial code vectors at the next iteration. This procedure continues until there is no significant change in the total distortion between cluster members and the code vectors around which they are clustered. Then the training vectors are compared with codebook that is generated by the LBG algorithm, to get the index position of the codeword with minimum distortion. This index value is transmitted to the receiver. The decoder at the receiver end has the same codebook as the encoder, and the decoding is performed by table look-up procedure using the received index value.

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 codewords in the codebook. Hence the encoding time complexity in the full search algorithm is given as KN operations where K is the input vector dimension, N is the codebook size per vector, and this complexity grows as the input vector dimension increases. To overcome this problem, a fast search algorithm is reported for VQ based image compression [13], which uses time reduction for searching the matched vector. In [14], a DCT based codebook generation method is reported that reduces the codebook size and search time. In this method, the features of training vectors are extracted by using DCT. By using the energy preserving property of the DCT the dimension of the feature vector is reduced. The reduced dimension feature vectors are used to design the binary tree codebook. In [15], a method is suggested to reduce the size of the codebook. In [16], Chou et al. have proposed an approach for designing an unbalanced Tree Structured Vector Quantization (TSVQ). The pruning algorithm used in this approach repeatedly removes the branch of the TSVQ and yields the smallest slope of increase in distortion to decrease in rate. The result is an unbalanced subtree that has the minimum average distortion with a

Dr. R. Krishnamoorthy is with Information Technology Department, Bharathidasan Institute of Technology, Anna University, Trichy – 620024, TamilNadu, India.(Phone: +91 9442587884; e-mail: rkrish26@hotmail.com).

N. Kannan is with Information Technology Department, Bharathidasan Institute of Technology, Anna University, Trichy – 620024, TamilNadu, India (e-mail: n_kannan_k@hotmail.com).

prescribed average rate. Moayeri et al. [17] propose a fine-coarse VQ that operates in two stages: a fine structured VQ followed by a coarse unstructured VQ using table look-up. In [18], a technique is proposed for encoding time reduction in vector quantization. Another efficient algorithm called Pairwise Nearest Neighbor (PNN) has been proposed by Equitz [19]. This algorithm requires no initial codebook. The basic idea of PNN is to successively merge the two closest clusters into another cluster represented by the centroids of the two clusters. It starts with the entire training set and iterates until the set of vectors is reduced to the desired size. The median value of the vectors is used for partitioning the training set. The computational efficiency of the PNN algorithm is achieved by a preprocessing operation that partitions the training vectors into a k-dimensional tree.

A binary tree is the simplest and most widely used structure of decision trees. It consists of a root node and set of terminal and nonterminal nodes. Each nonterminal node consists of two descendant nodes, but terminal node does not. In [20, 21] Binary tree classifier technique is reported. This technique deals with how a pattern vector is classified by passing it through the tree from the root to a leaf. In this method, there are three major tasks implemented for designing the binary tree classifier: (i) to setup the structure of an optimal binary tree, (ii) to choose the most effective feature subset at each nonterminal node, and (iii) to choose the decision principle used at each nonterminal node. The overall design criteria of the binary tree classifier are reported in [22]. These criteria include minimum probability of error, minimum number of nodes, minimum path length, minimum expected search time etc. In early years different tree search techniques are reported in the literature [23, 24, 25]. These search techniques require a large amount of computation time, memory with no guarantee for optimality. To overcome this, some other related works [26, 27] focus on the choice of features and/or decision principle under the constraint of a balanced tree structure. It has the advantage of relatively lower computational complexity and memory requirement. These tree structures can be effectively utilized to generate a binary tree codebook for the VQ.

Motivated by the fact that VQ in Transformed domain results in lesser computational complexity and reduced memory requirement, when combined with binary tree codebook we propose in this paper a binary tree codebook generation algorithm using the orthogonal polynomials based transform coding. The generating formula for the proposed transform coding is well established in image compression and other image processing techniques [28, 29,30]. The proposed transform coding has been configured as an integer transformation so as to reduce time and space complexities. The results show that the proposed binary tree codebook in VQ significantly reduces the computation time and obtains the better reconstructed picture quality compared to the PNN algorithm and DCT based VQ.

The proposed scheme is modeled as a variation of TSVQ (Tree Structured Vector Quantization). The primary difference between this scheme and the earlier works on TSVQ is the choice of split criterion. The earlier works makes use of the LBG at each node to split the training vectors into two partitions, whereas in our proposed scheme orthogonal polynomials based transform coefficients are used. Moreover the main computation required for growing a tree is just a scalar operation and no iteration is required.

This paper is organized as follows. In Sections II and III the proposed orthogonal polynomials based transform coding is presented. A brief introduction to Vector Quantization is presented in Section IV. In Section V, the proposed Codebook generation technique is described. The performance analysis measure used to evaluate the proposed technique is described in Section VI. Finally, the experiments and results are discussed in Section VII.

## II. Orthogonal Polynomials Based Transform Coding

A linear 2-D image formation system usually considered around a Cartesian coordinate separable, blurring, point spread operator in which the image I results in the superposition of the point source of impulse weighted by the value of the object f. Expressing the object function f in terms of derivatives of the image function I relative to its Cartesian coordinates is very useful for analyzing and compressing the image. The point spread function M(x, y) can be considered to be real valued function defined for (x, y) ∈ X x Y, where X and Y are ordered subsets of real values. In case of gray-level image of size (n x n) where X (rows) consists of a finite set, which for convenience can be labeled as {0, 1, …, n-1}, the function M(x, y) reduces to a sequence of functions.

$$M(i, t) = u_i(t), \; i = 0, 1, \ldots, n-1 \qquad (1)$$

The linear two dimensional transformation can be defined by the point spread operator M(x, y) (M(i, t) = ui(t)) as shown in equation 2.

$$\beta'(\zeta,\eta) = \int_{x\in X}\int_{y\in Y} M(\zeta,x)\, M(\eta,y)\, I(x,y)\, dxdy \qquad (2)$$

Considering both X and Y to be a finite set of values {0, 1, and 2 … n 1}, equation (2) can be written in matrix notation as follows

$$\left|\beta'_{ij}\right| = \left(\left|M\right|\otimes\left|M\right|\right)^{t}\left|I\right| \qquad (3)$$

where ⊗ is the outer product, $|\beta'_{ij}|$ are $n^2$ matrices arranged in the dictionary sequence, $|I|$ is the image , $|\beta'_{ij}|$ are the coefficients of transformation and the point spread operator |M| is

$$\left|M\right| = \begin{vmatrix} u_0(t_1) & u_1(t_1) & \cdots & u_{n-1}(t_1) \\ u_0(t_2) & u_1(t_2) & \cdots & u_{n-1}(t_2) \\ & \vdots & & \\ u_0(t_n) & u_1(t_n) & \cdots & u_{n-1}(t_n) \end{vmatrix} \qquad (4)$$

We consider the set of orthogonal polynomials u0(t), u1(t), …, un-1(t) of degrees 0, 1, 2, …, n-1, respectively to construct the polynomial operators of different sizes from equation (4) for n ≥ 2 and ti = i. The generating formula for the polynomials is as follows.

$$u_{i+1}(t)=(t-\mu)u_i(t)-b_i(n)u_{i-1}(t) \quad for\ i \geq 1, \qquad (5)$$

$$u_i(t)=t-\mu \quad and \quad u_0(t)=1$$

where

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

and

$$\mu=\frac{1}{n}\sum_{t=1}^{n}t$$

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

$$b_i(n)=\frac{i^2\left(n^2-i^2\right)}{4\left(4i^2-1\right)}, \qquad \mu=\frac{1}{n}\sum_{t=1}^{n}t=\frac{n+1}{2}$$

We can construct point-spread operators |M| of different size from equation (4) using the above orthogonal polynomials for n ≥ 2 and $t_i$ = i. For the convenience of point-spread operations, the elements of |M| are scaled to make them integers.

## III. The Orthogonal Polynomial Basis

For the sake of computational simplicity, the finite Cartesian coordinate set X, Y is labeled as {1, 2, and 3}. The point spread operator in equation (3) that defines the linear orthogonal transformation for image coding can be obtained as |M| $\otimes$ |M|, where |M| can be computed and scaled from equation (4) as follows.

$$|M|=\begin{vmatrix} u_0(x_0) & u_1(x_0) & u_2(x_0) \\ u_0(x_1) & u_1(x_1) & u_2(x_1) \\ u_0(x_2) & u_1(x_2) & u_2(x_2) \end{vmatrix}=\begin{vmatrix} 1 & -1 & 1 \\ 1 & 0 & -2 \\ 1 & 1 & 1 \end{vmatrix} \qquad (6)$$

The set of polynomial basis operators $o_{ij}^n\,(0\leq i, j\leq n-1)$ can be computed as $o_{ij}^n=\hat{u}_i\otimes\hat{u}_i^t$, where $\hat{u}_i$ is the (i + 1) st column vector of |M|.

The complete set of basis operators of sizes (2 X 2) and (3 X 3) are given below:

Polynomial basis operators of size (2 * 2) are

$$\left[O_{00}^{2}\right]=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}, \quad \left[O_{01}^{2}\right]=\begin{bmatrix}-1 & 1\\ -1 & 1\end{bmatrix},$$

$$\left[O_{10}^{2}\right]=\begin{bmatrix}-1 & -1\\ 1 & 1\end{bmatrix}, \quad \left[O_{11}^{2}\right]=\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}$$

Polynomial basis operators of (3 * 3) are

$$\left[O_{00}^{3}\right]=\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix},$$

$$\left[O_{01}^{3}\right]=\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix},$$

$$\left[O_{02}^{3}\right]=\begin{bmatrix}1 & -2 & 1\\ 1 & -2 & 1\\ 1 & -2 & 1\end{bmatrix},$$

$$\left[O_{10}^{3}\right]=\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{bmatrix},$$

$$\left[O_{11}^{3}\right]=\begin{bmatrix}1 & 0 & -1\\ 0 & 0 & 0\\ -1 & 0 & 1\end{bmatrix},$$

$$\left[O_{12}^{3}\right]=\begin{bmatrix}-1 & 2 & -1\\ 0 & 0 & 0\\ 1 & -2 & 1\end{bmatrix},$$

$$\left[O_{20}^{3}\right]=\begin{bmatrix}1 & 1 & 1\\ -2 & -2 & -2\\ 1 & 1 & 1\end{bmatrix},$$

$$\left[O_{21}^{3}\right]=\begin{bmatrix}-1 & 0 & 1\\ 2 & 0 & -2\\ -1 & 0 & 1\end{bmatrix},$$

$$\left[O_{22}^{3}\right]=\begin{bmatrix}1 & -2 & 1\\ -2 & 4 & -2\\ 1 & -2 & 1\end{bmatrix}$$

It can be shown that a set of (n x n) (n ≥ 2) polynomial operators forms a basis, i.e. it is complete and linearly independent. In the next section, we present first a brief introduction on Vector Quantization and then the proposed Codebook generation algorithm.

## IV. Vector Quantization (VQ)

A vector quantizer is a system for mapping a sequence of continuous or discrete vectors into a digital sequence suitable for storage. The main reason for doing this is to reduce the rate for transmission of an image.

### A. *VQ Definition*

A vector quantizer Q of dimension k and size N is a mapping from a point in k-dimensional Euclidean space, $R^k$ into a finite set C containing N output or reproduction points that exist in the same Euclidean space as the original point. These reproduction points are known as code words and these set of code words are called a codebook C with N distinct code words in the set. Thus, the mapping function Q is defined as,

$$Q : R^k \rightarrow C, \tag{7}$$

The rate of the vector quantizer or the number of bits used to express each quantized vector is,

$$r = \log_2 N / k \tag{8}$$

This rate equation is very useful as it gives the amount of compression that can be expected for a particular VQ coding scheme. Vector quantization in its entirety is quite a simple concept, but the major complexity comes about in selecting a codebook C of size N that best represents original vectors or training set X in Rk Euclidean space. To solve this optimization problem, we require a distortion measure $d(X,\hat{X})$ that represents the penalty of the mapping Q process. The measure is given by $d(X,\hat{X}) = \sum_{j=0}^{k-1}(x_j - \hat{x}_j)^2$, where $x_j$ *ana* $\hat{x}_j$ are jth elements of X (training vector) and $\hat{X}$ (codebook vector) vectors, respectively.

### B. *Vector Quantizer Design*

The goal to design an optimal vector quantizer is to obtain a quantizer consisting of N reproduction vectors, such that it minimizes the expected distortion. The codebook is designed from the set of training vectors and a popular design method used is LBG algorithm. It is an iterative algorithm that requires an initial codebook, which is obtained by splitting method. In this method an initial code vector is set as the average of the entire training sequence, which is defined by the source image. This code vector is then split into two. These two code vectors are split into four and is repeated until the desired number of code vectors is obtained.

In the encoding process, the input vector simply searches for the best matching codebook vector in the obtained codebook and the index of that vector in a codebook is transmitted. The decoder performs as the table look-up. It receives the transmitted index and look at the codebook for the vector that corresponds to and then matched vector from the codebook is used to represent the input vector.

## V. Proposed Codebook generation

In this section, a vector quantization scheme that facilitates image compression is presented with orthogonal polynomials based transformation. We consider this as a classification problem since in this scheme the codebook is generated by partitioning the feature patterns into clusters. The proposed technique uses two steps namely extraction of features using orthogonal polynomials based transform coding and design of binary tree classifier. The advantage of combining both the transformation and VQ using the proposed binary tree classifier codebook is that, when a linear transform is applied to the vector signal, the information is compacted into a subset of the vector components. In the frequency 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. These low energy components might be discarded entirely. Let us consider a k-dimensional transformed input vector, Y. We map it into a d-dimensional vector (d < k), Z by discarding the low frequency components. Then the p-dimensional VQ is used to vector quantize the truncated vector, forming the quantized approximation $\hat{Z}$. Recovery can be obtained by padding operation that appends (k–d) additional components with value zero to the vector, producing the k-dimensional vector $\hat{Y}$. Finally the inverse transform is applied to $\hat{Y}$ to obtain the approximate input image vector.

The features are extracted from frequency domain coefficients that are obtained from the orthogonal polynomials based transform coding of the original image data, and with reduced dimension of data considerably thereby reducing the computational time. Then, a binary tree classifier is proposed to generate codewords from the extracted features. The binary tree classifier focuses 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.

In this proposed scheme, the binary tree classifier uses the feature with the largest variance as the split key. 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. Now that the split key has been chosen, the decision principle has to be decided upon, which is a simple threshold operation. Here the mean value of the corresponding split key feature is used as the threshold because the mean not only contains most of the statistical information of the feature but also requires less computation. A threshold T is used to partition the training feature vectors at a nonterminal node into two halves. The partition is based on whether the key value of the vector is greater or lesser than T. 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 codewords of the codebook are formed by computing the centroids of the feature vectors falling at each node.

The following example illustrates the proposed codebook design using the binary tree classifier. Assuming that we are interested to generate a codebook with a size of 4 from the training set {x1, x2...... x9}. All the training vectors are partitioned by the key k1 into two groups: G1 = {x1, x2, x6, x9} and G2 = {x3, x4, x5, x7, x8}. G1 is further partitioned by K2 into G3 = {x1, x6} and G4 = {x2, x9}; G2 is partitioned by k3 into G5 = {x3, x7} and G6 = {x4, x5, x8}. The mean of the training vectors falling at each terminal node is calculated and regarded as code vector of the codebook.

The proposed codebook generation algorithm using the binary tree classifier and Orthogonal Polynomials based transform coding is described hereunder.

## A. Codebook generation algorithm

Input: Image of size N x N
Output: Binary tree codebook

Step 1:
Partition the training image into non-overlapping blocks of size M * M (typically M = 4). A block of pixels is called a training pattern vector and denoted as x, x = {x (j), j = 1, 2… k; k = M * M}.

Step 2:
Extract the features of each block from the orthogonal polynomials based transform coding as described in section 2.

Step 3:
Rearrange the feature coefficient matrix into 1-D array in zig-zag sequence in the order of decreasing variance to form k-dimensional feature vector.

Step 4:
Discard the high frequency coefficients based on energy preserving property of the proposed transform coding to form the feature vector z, where z = {z(j), j = 1,2,…,d; d < k}.

Step 5:
Compute the mean and variance of each feature. For any nonterminal node n, the mean Mn (j) and variance Vn (j) for the jth feature are defined as

$$M_n(j) = N_n^{-1} \sum_{i=1}^{N_n} Z_i^n(j) \qquad (9)$$

$$V_n(j) = \sum \left[ Z_i^n(j) - M_n(j) \right]^2 \qquad (10)$$

for j = 1, 2…, k where Nn is the number of training vectors at the node n and $Z_i^n(j)$ is the jth feature corresponding to training vectors at node n.

Step 6:
Select the feature with the largest variance as the split feature and choose the corresponding mean value as the threshold. If Vn (p) = max {Vn (j), j = 1,2,…,d}, then the pth feature of the feature space will be chosen as a split feature at this node, and Mn (p) is adopted as a split threshold.

Step 7:
Compare the value of the pth feature of the training vector with Mn (p). If it is less place the pth feature on the left sub tree; otherwise on the right sub tree.

Step 8:
Repeat steps 2 through 7 until the required number of clusters are formed.

Step 9:
Compute the centroid of the feature vectors falling on each of the terminal node of the binary tree classifier and use it as code vector of the final codebook.

This proposed codebook design is used to vector quantize the transform coefficients. After vector quantizing, the quantizer outputs are inverse transformed using the basis functions as described in Section 3 to produce the quantized approximation of the original input image.

## VI. Performance Analysis

The performance of the proposed vector quantizer design on orthogonal polynomials based transform coding is reported by computing the peak signal-to-noise ratio (PSNR), which is defined as

$$PSNR = 10\log_{10}\left[\frac{255}{e_{ms}^{\;2}}\right]^2 \qquad (11)$$

where the average mean-square error ems, is

$$e_{rms}^2 = \frac{1}{NM}\sum_{i=1}^{N}\sum_{i=1}^{M} E\left(u_{i,j} - u'_{i,j}\right)^2$$

where $\{u_{i,j}\}$ and $\{u'_{i,j}\}$ represent the N x M original and reproduced images respectively.

## VII. Experiments & Results

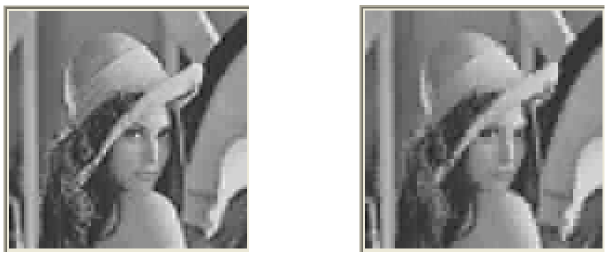
The proposed codebook design for Vector Quantization scheme has been implemented on various test images. One such test image viz. Lena image of size 128 x 128 with pixel values in the range 0 – 255 is given in figure 1. The input image is partitioned into various non-overlapping sub-images of equal size of (4 x 4) and the codebook size is 128. This block is subjected to orthogonal polynomials based transform coding and features are obtained as explained in section 2. The binary tree classifier is constructed using the energy preserving features as described in section 5 and VQ is implemented. The image is reconstructed using the orthogonal polynomial basis functions as described in section 3. The performance of the proposed vector quantization scheme is measured with Peak-Signal-to-Noise-Ratio (PSNR) as given in equation (11). We could achieve a Mean Square Error value of 32.19 with a PSNR value 33.05dB for the input image

Lena. The reconstructed image by the proposed scheme corresponding to the given original image is shown in figure 2(a). This proposed method is also compared with PNN method and DCT based binary tree codebook method. The reconstructed images by these two methods corresponding to the original images are shown in figure 2(b) and 2(c) respectively. The Mean Square Error with PSNR values by the proposed method, PNN method and DCT based binary tree codebook method are presented in Table 1. From the experimental results, it is observed that the proposed coding gives higher PSNR value with good reconstructed image quality.

TABLE I
PERFORMANCE COMPARISON

| CODING SCHEME | MSE | PSNR(DB) | BITS/PIXEL |
|---|---|---|---|
| PNN | 35.14 | 32.67 | 0.25 |
| VQ USING DCT | 33.49 | 32.88 | 0.25 |
| PROPOSED SCHEME | 32.19 | 33.05 | 0.25 |

Fig. 1 Original image


(a) Proposed Scheme (b) PNN scheme

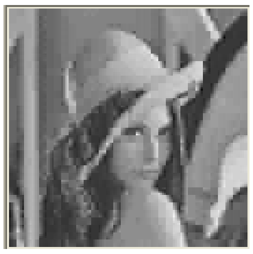

(c) VQ using DCT

Fig. 2 Reconstructed images

## VIII. CONCLUSION

In this paper a new codebook design algorithm is proposed for the vector quantization of images. This scheme uses orthogonal polynomials based transformation to extract the features of the training images. Using the energy preserving property of the proposed transformation, certain significant components of the feature space are selected to partition the training vectors into a binary tree. A single feature with the largest variance and its corresponding mean value are adopted as split feature and split threshold respectively for generating the binary tree codebook. As a result, the binary tree codebook is used to vector quantize the image under analysis. The performance of the proposed scheme is measured with standard PSNR value and is compared with PNN method and VQ using DCT.

## REFERENCES

[1] Jain A.K., “Image Data Compression: A Review,” Proc. IEEE, Vol. 69, No. 3, pp. 349 – 389, 1981.
[2] G.K. Wallace, “The JPEG Still Picture Compression Standard,” Communications of ACM Vol. 34, pp.31-44, 1991.
[3] N. Ahamed, T. Natarajan and K.R. Rao, “Discrete Cosine Transform,” IEEE Transactions on Computer, Vol. 23, pp. 90-93, 1974.
[4] R.C. Reininger and J. Gibson, “Distribution of 2-Dimensional DCT Coefficients for Images,” IEEE Transactions on Communications. Vol. 31, pp. 835-839, 1983.
[5] H.C. Andrews, “Computer Techniques in Image Processing,” Academic Press, New York, 1970.
[6] W.K. Pratt, W.H. Chen and L.R. Welch, “Slant Transform Image Coding,” IEEE Transactions on Communications, Vol. 22, pp. 1075-1093, 1973.
[7] A. Habibi and P.A. Wintz, “Image Coding by Linear Transformation and Block Quantization,” IEEE Transactions on Communications Technology. Vol.19, pp. 50-62, 1971.
[8] P.A. Wintz and M.Tasto, “Image Coding by Adaptive Block Quantization,” IEEE Transactions on Communications Technology, Vol. 19, pp. 957-972, 1971.
[9] Whelche. J.E, Jr., and Guinn, D.F., “The Fast Fourier Hadamard Transform and its Use in Signal Representation and Classification,” Eascon 1968 convention record, pp. 561-573, 1988.
[10] Henderson. K.W, “Some Notes on Walsh Functions,” IEEE Transactions on Electronic Computers. Vol. EC-13, No.1, pp. 50-52, 1964.
[11] Y. Linde, A. Buzo and R.M. Gray, “An algorithm for Vector Quantizer Design,” IEEE Transactions on Communications, Vol. 28, pp 84-95, January 1980.
[12] R.M.Gray, “Vector Quantization”, IEEE ASSP Magazine, pp. 4-29, April 1984.
[13] Hsieh.C.H. , Lu.P.C. and Chung. J.C, “Fast Codebook Generation Algorithm for Vector Quantization of Images”, Pattern Recognition Letter, 12, pp. 605-609. 1991.
[14] Hsieh. C.H., “DCT based Codebook Design for Vector Quantization of Images”, IEEE Transactions on Circuits and Systems, Vol. 2, pp. 401-409, 1992.
[15] Sangeetha Ramakrishnan, Kenneth Rose and Alen Gersho, “Constrained Storage Vector Quantization with a Universal Codebook”, IEEE Transactions on Image Processing, Vol. 7, No. 6, pp. 785 - 793, June 1998.
[16] P.A.Chou, T.Lookabaugh, and R.M.Gray, “Optimal pruning with applications to tree-structured source coding and modeling,” IEEE Transactions on Information Theory, Vol. 35, pp. 299 – 316,1989.
[17] [17] N.Moayeri, D.L.Neuhoff, and W.E.Stark, “Fine-Coarse vector quantization,” IEEE Transactions on ASSP, Vol. 39, pp. 1503-1515, 1991.
[18] Peter Vprek and A.B. Bradley, “An improved algorithm for Vector Quantizer Design”, IEEE Signal Processing Letters, Vol. 7, No. 9, pp 250-252, September 2000.

[19] W.Equitz "A new vector quantization clustering algorithm", IEEE Transactions on ASSP, vol. 37, pp.1568-1575, Oct. 1989.
[20] [20] S.Q.Yun and K.S.Fu, "A method for design of binary tree classifiers," Pattern Recognition, Vol. 16, pp. 509-603, 1983.
[21] Y.K.Lin and K.S.Fu, "Automatic classification of cervical cells using a binary tree classifier," Pattren Recognition, vol. 16, pp. 69-80, 1983.
[22] X.Li and R.C.Dubes , "Tree classifier design with a permutation statistic," Pattern Recognition, Vol. 19, pp. 229-235, 1986.
[23] [23] P.Swain and W.Meisel, "The decision tree classifier design and potential," IEEE Transactions on Geosci. Electron., Vol. 15, pp. 142-147, 1977.
[24] A.V.Kulkarni and L.N.Kanal, "An optimization approach to hierarchical classifier design," in Proceedings Third International joint Conference on Pattern Recognition, pp.459-466, 1976.
[25] H.J.Payne and W.S.Meisel , "An algorithm for constructing optimal binary decision trees," IEEE Transactions on Computers, Vol. 26, pp. 905-916, 1977.
[26] M.W.Kurrzynski, "The optimal strategy of a tree classifier," Pattern Recognition, Vol. 16, pp. 81-87, 1983.
[27] A.V.Kulkarni, "On the mean accuracy of hierarchical classifiers," IEEE Transasctions on Computers, Vol. 27, pp. 771-776, 1978.
[28] R. Krishnamoorthi and P. Bhattacharayya, "A new data compression Scheme using Orthogonal Polynomials", IEEE Proceedings on International Conference on Information, Communication and Signal Processing, Nanyang Technology University, Singapore, Vol-I, pp-490-494, 1997.
[29] R.Krishnamoorthi "Transform coding of monochrome images with a statistical design of experiments approach to separate noise" Pattern Recognition Letters Vol.28, pp. 771-777. 2007.
[30] R.Krishnamoorthi, "A unified framework orthogonal polynomials for Edge detection, Texture Analysis and Compression incolor images," Ph.D. Thesis, I.I.T, Kharagpur, 1998.