Fast vector Quantization of Color Image Coding with Single Codebook based on Orthogonal Polynomials

Krisshnamoorthy R
Computer Vision Lab, Department of CSE,
Anna University of Technology, Tiruchirappalli – 620 024.

Punidha R
Computer Vision Lab, Department of CSE,
Anna University of Technology, Tiruchirappalli – 620 024.

ABSTRACT

In this paper, a new fast vector quantization encoding technique for transform coding of RGB color images that does not require a color coordinate conversion matrix is proposed. The proposed work directly applies the orthogonal polynomials transformation on the input image and transformed training set with reduced dimension is obtained for the vector quantization and hence the proposed work has reduced computational complexity. In the codebook generation phase of vector quantization encoding, a new transformed binary tree algorithm is proposed to construct 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 and so a big saving in codebook construction time is achieved. A new transformed tree structured codeword matching algorithm is proposed in order to further reduce the vector quantization encoding time for finding the closest codeword of an input vector. The experimental results show that the proposed algorithm greatly reduces the encoding time when compared with recent fast vector quantization algorithms.

General Terms

Image Coding, Vector Quantization.

Keywords

Orthogonal Polynomials Transform, Transformed Binary Tree, Transformed Tree Structured Codeword Matching.

1. INTRODUCTION

We In the current scenario, Vector Quantization (VQ) is found to be an efficient technique for data compression and is used in numerous applications including image encoding and image recognition [6]. Various type of VQ techniques such as Address VQ [13], Classified VQ [3], Finite state VQ [9], Adaptive VQ [7] have been reported for various purposes. Vector quantization based image compression [4,14,15] 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 is reported by combining both transform coding and VQ that takes the 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 phase and codeword matching phase. For the codebook generation technique, the Linde–Buzo-Gray (LBG) algorithm [17], which is also called the Generalized Lloyd Algorithm (GLA) is the most commonly technique. Techniques for fast and efficient codebook generation of vector quantization have been reported in the literature. Various improvements [5, 8, 11] for LBG based codebook construction algorithm have been adopted to minimize the computing time and to generate better codebook for representing the input vector. Most of the existing vector quantization algorithms are experimented with monochrome images and few vector quantization techniques have been reported with color image coding. A color image coding technique with variable blocks size is reported in [12], wherein the DCT transform is combined with vector quantization. A wavelet based color image compression is implemented in [2], in which the RGB color space is converted into YUV color space and the vector quantization is applied as if there are three monochrome images.

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, generally the vector quantization technique employs the full search algorithm, and examines the euclidean distance between the input vector and all the codewords 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 [1, 10, 18]. In [10], an algorithm for fast codeword search is presented wherein the Karhunen-Loeve Transform (KLT) transformed codewords and partial distortion rejection is used to find the closest codeword. Recently, a fast codeword search algorithm is reported in [1] and uses projection pyramids and hadamard transform to eliminate unlikely codewords,

In essence, both the VQ encoding phase viz., codebook generation and codeword matching algorithm should be optimized in terms of computation time. In this proposed work, a binary tree codebook is constructed with the proposed orthogonal polynomials transformed training vectors. Since the proposed transformation 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 transformation is not required in the proposed scheme. The transformed binary tree codebook algorithm has the advantage that the computational complexity grows linearly with rate, rather than exponentially as in GLA. Once the codebook is constructed using the proposed Transform Binary Tree (TBT) algorithm, the second phase of VQ encoding involves closest codeword search algorithm. For efficient codeword search, a new algorithm called Transformed Tree structured Codeword Matching (TTCM) is proposed, which traverses minimum number of nodes to find the best matched codeword.

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 orthogonal polynomials transform based VQ encoding (FOPTVQE) is presented in section 4. The experimental results and comparison with existing techniques are presented in section 5 and the conclusion is presented in section 6.

2. ORTHOGONAL POLYNOMIALS BASED TRANSFORM CODING

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

$$I(x,y,z)=\int\int\int f(\xi,\eta,\gamma)\,d\mu(\xi)d\mu(\eta)d\mu(\gamma) \quad (1)$$

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 [16], 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
$$f(x)=\int\delta(x-t)I(t)dt \quad (2)$$
2. The ordinary linear transformation
$$f_i=\sum_{k=0}^{n-1}u_{ik}I_k,(i=0,...,n-1) \quad (3)$$

The linear transformation defined in equation (2) by the matrix $|\delta(x_i-t_i)|$ is a smoothing operation provided it is totally positive, whereas the linear transformation defined in equation (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

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

As shown in equation (2), the process of image analysis can be viewed as the linear transformation defined by the point-spread operator *M(x, y) (M(i, t) = u$_i$(t))*

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

where $\zeta,\eta,$ *and* s are coordinates in the 3-D transformed space and $I(x,y,z)$ is a color image region wherein *x* and *y* are two spatial coordinates and *z* indicates the color space. Considering each of *X, Y* and *Z* to be finite set of values {0, 1, … *n*-1} equation (2) can be written in matrix notation as follows

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

where the point spread operator $|M|$ is

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

$\otimes$ 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.

$$u_{i+1}(x)=(x-\mu)u_i(x)-b_i(n)u_{i-1}(x)\quad for\ i\geq 1, \quad (7)$$

$$u_1(x)=x-\mu,\quad and\quad u_0(x)=1, \quad (2)$$

where

$$b_i(n)=\frac{\langle u_i,u_i\rangle}{\langle u_{i-1},u_{i-1}\rangle}=\frac{\sum_{x=1}^n u_i^2(x)}{\sum_{x=1}^n u_{i-1}^2(x)}\quad \text{and} \quad (3)$$

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

Considering the range of values of x to be $x=i,\quad i=1,2,3,...,n,$ we get

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

Next, we construct point-spread operators |*M*|s of different width from the above orthogonal polynomials using equation (5).

3. 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 equation (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 equation (7) as follows.

$$|M|=\begin{vmatrix} u_0(x_0) & u_1(x_0) & u_2(x_0) \\ u_0(x_1) & u_1(x_1) & u_2(x_1) \\ u_0(x_2) & u_1(x_2) & u_2(x_2) \end{vmatrix} = \begin{vmatrix} 1 & -1 & 1 \\ 1 & 0 & -2 \\ 1 & 1 & 1 \end{vmatrix} \quad (8)$$

$$O_{ijk} = \hat{u}_i \otimes \hat{u}_j \otimes \hat{u}_k$$

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 equation (5) Having described the Orthogonal Polynomials model and its basis, we present the concept of proposed fast vector quantization in the next section.

4. PROPOSED CODEBOOK GENERATION AND CODEWORD MATCHING TECHNIQUE

In this section, the proposed VQ encoding algorithm called Fast Orthogonal Polynomials Transform based VQ Encoding (*FOPTVQE*) is presented. In the literature of transform based color image coding systems, a preprocessing step is done as the color coordinate conversion before the application of orthogonal transformation in order to reduce the correlation between R, G and B color planes. Then the new color coordinate systems are encoded as if they were three monochrome images using any one of the transform based coder such as DFT, DCT, DWT, and KLT. The inter-correlation of individual color planes and interactions among the color planes is not effectively utilized in the exiting color image coding systems and the color coordinate conversion is an additional overhead. In this proposed work, new orthogonal polynomials based color image transform coding technique is proposed that does not require color coordinate conversion process for compression of composite color images. The proposed algorithm consists of three steps: extraction of features with orthogonal polynomials transform coefficients, design of Transformed Binary Tree (*TBT*) for codebook generation and Transformed Tree Structured Code word Matching(*TTCM*). The input image is divided into ($n \times n$) non overlapping region and then the unitary orthogonal polynomials transform is applied to extract the frequency features β'_{ijk} with dimension *k* on the combined three color image regions. Being unitary, the high energy components in the transformed domain are concentrated in the low frequency region. These low energy components are discarded and transformed training vectors with reduced dimension *d* are obtained for vector quantization. From the *d*-dimensional transformed training vectors, the binary tree based algorithm is used for codebook generation since it reduces both codebook generation time and codeword matching of VQ encoding algorithm.

The Transformed Binary Tree(*TBT*) 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 (*T*) because the mean not only contains most of the statistical information of the feature but also requires less computation. The threshold *T* 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 the 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.

(8)

In the code word matching of VQ encoding, the binary tree based search algorithm called Transformed Tree structured Codeword Matching (*TTCM*) algorithm with a better time-quality trade-off to obtain the closest codeword is implemented in *FOPTVQE* algorithm because Full Search (FS) technique is an exhaustive search approach and is time consuming. In the proposed algorithm, the reduced *d*-dimensional codebook binary tree is dynamically traversed with minimum number of nodes to find closest codeword index *I* of input training vector *x*. Initially, the algorithm is proposed to find the local closest codeword with single path tree search method. Next it limits the number of search paths in the binary tree using critical function as given below.

$$\mathrm{F}(x,n)=\frac{|MSE(x,child_L(n))-MSE(x,child_R(n))|}{MSE(x,child_L(n)+MSE(x,child_R(n))} \quad (9)$$

where *x* is input training vector, *n* is non terminal node, $child_L(n)$ and $child_R(n)$ denote the left and right child node on *n* respectively. *MSE ( )* is the mean square error between the two vectors. The threshold value *TH* is used along with critical function to determine dynamically whether to add paths for further search. If $F(x,n) \le TH$ then the algorithm selects both children of node *n* to search, otherwise it only traverses the nearest child node to *x*. Finally the algorithm finds the global closest codeword index c_i and its index *I*, which is further compressed using entropy coding, to result in compressed data stream. 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. 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 block diagram of the proposed color image coding

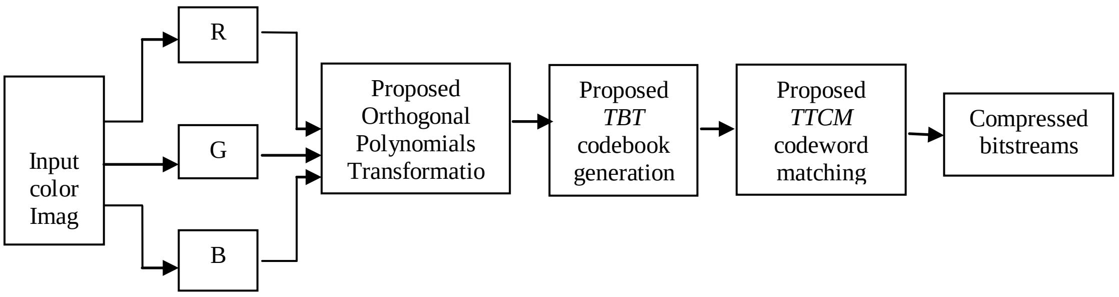

Input color Imag
R
G
B
Proposed Orthogonal Polynomials Transformatio
Proposed TBT codebook generation
Proposed TTCM codeword matching
Compressed bitstreams

Figure 1 Block diagram of proposed *FOPTVQE* color image coding technique

algorithm is given in fig.1. The proposed Fast Orthogonal Polynomials Transform based Vector Quantization Encoding (*FOPTVQE*) technique is presented as an algorithm hereunder.

The proposed *FOPTVQE* Algorithm:

Input : Color image of size (*image_width*$\times$*image_height) and* 3-Dimensional polynomials operator

$$|\mathcal{M}| = |M| \otimes |M| \otimes |M|$$

Output : Encoded color image with proposed *FOPTVQE* algorithm.

Step:

```
1. Initialize blks=(image_width × image_height) / (n × n).
2. Partition the input image into non-overlapping blocks of size (n × n) and obtain input block of pixel values [I] from three individual R, G and B color components for total number of blocks blks.
3. Compute the orthogonal polynomials transformed co-efficients $\beta'_{ijk} = [\mathcal{M}]^t [I]$ with dimension k as described in section II.
4. Discard the high frequency coefficients based on energy preserving property of the proposed transform coding and obtain the transformed training vectors X={xi, i=1,2,...blks} with dimension d for all the blocks as described in section V.
5. Apply the proposed TBT algorithm as described below and construct the codebook C with N code words and each codeword consists of d-dimensional code vector represented as $c_{ij}$ ( i=1 to N, j=1,2,..,d, where d < k).
6. For each input transformed training vector x in X, apply the proposed TTCM algorithm to find the closest code word index I from the codebook C.
7. Encode the closest codeword index I using entropy encoding for further compression.
END.
```

Proposed Transformed Binary Tree(*TBT*) codebook generation algorithm:

Input : Transformed training vectors X each with d-dimensional features.

Output : Codebook C with N codeword with proposed algorithm.

Step:

```
1. Partition all the input training vectors X into a binary tree.
   (a) Compute the mean and variance of each feature from d-dimensional transformed training vectors X. For any non terminal node n, the mean $M_n(j)$ and variance $V_n(j)$ for the $j^{th}$ feature are defined as
       $$M_n(j) = \frac{1}{X_n} \sum_{i=1}^{X_n} z_i^n(j)$$
       $$Var_n(j) = \frac{1}{X_n} \sum_{i=1}^{X_n} \left[ z_i^n(j) - M_n(j) \right]^2$$
       where j = 1, 2..., d and $z_i^n(j)$ is the $j^{th}$ feature corresponding to training vectors at node n.
   (b) Select the feature with the largest variance
       $V_n(p) = \max\{V_n(j),\ j = 1,2,...,d\}$
       as the split feature and the corresponding mean value $M_n(p)$ as the threshold T.
   (c) Compare the value of the $p^{th}$ feature of the training vector with split threshold. If it is less than $M_n(p)$, then place the vector on the left sub tree; otherwise place it on the right sub tree.
   (d) Repeat (a) through (c) until the required number of clusters is formed.
2. Compute the centroid of the feature vectors falling on each of the terminal node of the binary tree and use it as code vector of the final codebook C.
END.
```

Proposed Transformed Tree structured Codeword Matching (*TTCM*) closest codeword search algorithm:

Input : Codebook C and input transformed training vector x.

Output : Closest codeword index I with proposed *TTCM* algorithm.

Step :

```
1. Read the input transformed training vector x and set root node of binary tree as first non terminal node n.
```

2. Find the distortion between *x* and left, right child of non terminal node *n* as defined in section IV.
3. Push the child node with greater distortion into the stack and set the other child node one as the current non terminal node *n*.
4. Repeat steps 2 and 3 until *n* is a leaf node.
5. Set the leaf node index *I* as local closest code word *LC*.
6. Pop up the non terminal node *n* from stack to search for other paths in the binary tree for finding global closest code word *GC*.
7. Calculate the critical function *F* using equation (9) to determine dynamically to whether add the paths for further search with user defined threshold *TH*.
8. If $F(x,n) \leq TH$, then select both the children of node *n* for further search and push the child node with greater distortion onto stack; otherwise traverse the child node with minimum distortion.
9. Repeat steps 6 through 8 until *n* is a leaf node.
10. Set the global closest code word *GC* as leaf node and output its index *I*.

END.

5. EXPERIMENTS AND RESULTS

The proposed *FOPTVQE* algorithm for color image coding has been experimented in Intel Core(2)-Quad, 2.3 GHZ speed processor system with 2000 test images. Two sample standard test images viz. lena and sailboat 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 figure 2(a) and 2(b) respectively. The input color images are partitioned into non-overlapping blocks 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 *X* are obtained for codebook training. The codebook is constructed with the proposed *TBT* algorithm for different sizes as described in section 4 and the computation time are noted in mille seconds(ms) for various test images including the sample images shown in figure 1. To construct the codebook of size 2048, the proposed algorithm took 92.124ms and 89.623ms for lena and sailboat images respectively. Similarly, for the codebook of size 512, the proposed algorithm took 63.340ms for lena image and 60.180ms for sailboat image. In the same way, when the codebook size is 128, the proposed *TBT* algorithm takes 47.118ms and 45.172ms for the same standard test images. The experiment is repeated for various code book sizes with different images and the results obtained by the proposed algorithm are presented in table 1. In order to measure the performance of the proposed codebook generation algorithm, we conduct experiments with recent fast codebook generation algorithms such as Pattern Reduction Enhanced GLA (PREGLA)[5] and Codebook Generation Algorithm Using Codeword Displacement (CGAUCD)[11] techniques on the same computer system and the results obtained are incorporated in the same table 1.

Figure 2. Original test images

The code word matching of VQ encoding involves in finding the index of best codeword that represents the input vector. The *FOPTVQE* technique uses Transformed Tree structure Codeword Matching (*TTCM*) algorithm to find closest codeword of input vector *x*. In the proposed *TTCM* algorithm, the threshold value is set to 0.5 to find the global closest code word *GC* with index *I*. This index *I* is further encoded using entropy coding to yield compressed bit streams. The computation times of codeword search using proposed *TTCM* algorithm for various codebook sizes are presented in table 2. For code word matching, when the code book size is 2048, the proposed *TTCM* algorithm takes 23.180ms and 21.543ms for lena and sailboat images respectively. When the codebook size is 512, the average execution time of the proposed algorithm is 11.520ms and 13.849s for lena and sailboat images. Similarly for a codebook of size 128, the average execution time of the proposed algorithm is 6.485s and 6.135ms for the same standard images. The performance of this proposed algorithm is also compared with existing fast codeword search algorithm Hadamard Projection Pyramid Search (HPPS)[1] and the results are included in table 2.

The performance of the proposed vector quantization is also measured with Peak-Signal-to-Noise-Ratio (PSNR) in decibels (dB) for various bits per pixel (bpp). For this measure, the decompression is done by first entropy decoding followed by inverse vector quantization as a simple table look up process in the codebook. Then the inverse transform is applied on the de-quantized coefficients with orthogonal polynomials basis as described in section 3 and the reconstructed color image is obtained. . A graph is plotted with bits per pixel in x-axis and the PSNR in the y-axis for the proposed and recent fast vector quantization encoding techniques and the same is presented in fig 3 for lena image. The result of reconstructed images corresponding to the original images shown in figure 2 with the proposed algorithm is presented in figure 3(a) and 3(b) for the codebook of size 1024. From the experimental results it is evident that the proposed fast vector quantization encoding technique with orthogonal polynomials transform greatly reduces the encoding time and gives good PSNR when compared with existing fast vector quantization techniques.

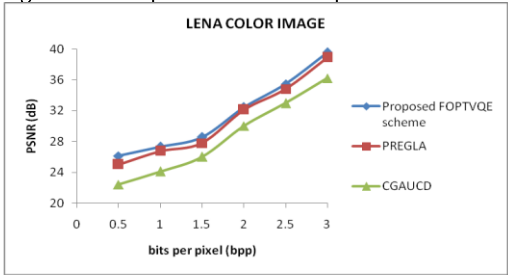
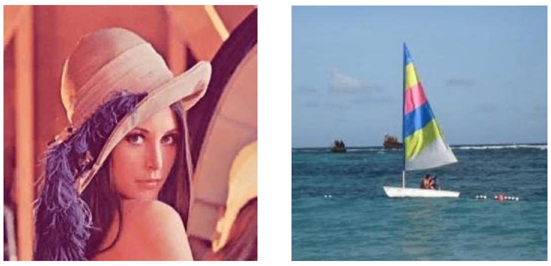


Figure. 3 bits per pixel(bpp) versus PSNR for the proposed *FOPTVQE*, PREGLA and CGAUCD schemes.



(a) (b)

Figure. 4. Results of proposed *FOPTVQE* technique when the code book size is 1024.

Table 1 Comparison of computing time (ms) for codebook generation using proposed *TBT*, PREGLA and CCAUCD schemes for set of training images.

Codebook size	**Method**	**Computation Time**				
		Lena	Sailboat	Peppers	Mandrill	Parrot
2048	Proposed *TBT*	92.124	89.623	91.020	92.518	93.290
	PREGLA	829.145	703.412	834.108	794.639	824.219
	CGAUCD	953.134	831.422	921.391	893.422	920.172
1024	Proposed *TBT*	78.578	75.190	76.152	78.378	77.304
	PREGLA	447.136	397.115	479.740	451.481	431.220
	CGAUCD	551.023	461.474	582.421	528.453	580.235
512	Proposed *TBT*	63.340	60.180	62.185	63.018	64.482
	PREGLA	366.145	302.049	375.271	389.569	342.790
	CGAUCD	451.859	398.906	463.891	458.922	471.266
256	Proposed *TBT*	55.239	53.111	57.687	57.392	56.470
	PREGLA	267.193	224.418	281.629	295.290	273.380
	CGAUCD	341.227	303.641	352.875	341.890	361.765
128	Proposed *TBT*	47.118	45.172	48.027	47.810	48.290
	PREGLA	202.140	198.331	211.591	221.204	226.510
	CGAUCD	286.330	249.500	296.359	291.125	280.390

Table 2 Computing time (ms) for codeword matching with proposed *TTCM* and HPPS technique for set of training images.

Codebook size	**Method**	**Computation Time**				
		Lena	Sailboat	Peppers	Mandrill	Parrot
2048	Proposed *TTCM*	23.180	21.543	28.542	27.452	26.070
	HPPS	332.165	324.701	342.432	339.317	331.325
1024	Proposed *TTCM*	17.331	18.413	21.213	20.197	20.014
	HPPS	250.309	235.590	266.378	250.101	234.610
512	Proposed *TTCM*	11.520	13.849	15.864	14.531	12.401
	HPPS	140.080	157.225	156.525	141.341	141.045
256	Proposed *TTCM*	9.780	8.413	11.115	10.321	9.420
	HPPS	94.319	93.314	99.779	97.039	78.020
128	Proposed *TTCM*	6.485	6.135	7.003	7.491	6.010
	HPPS	47.312	62.413	63.698	62.202	47.790

6. CONCLUSION

A new fast VQ encoding algorithm for color image transform coding is presented in this paper with orthogonal polynomials transformed reduced dimension training set. Due to the inter-correlation property of the proposed transform, a single codebook for vector quantization is generated for all the three color components of training image and hence the codebook construction time is reduced one third of existing techniques. The proposed transformed binary tree algorithm significantly reduces the codebook construction time when compared with recent fast codebook generation algorithms. The codeword matching phase of proposed vector quantization technique uses transformed tree structured codeword matching algorithm to find the closest codeword which also reduces the computation time total vector quantization encoding time. From the experiment results, it is shown that the proposed algorithm greatly reduces the computation time of total vector quantization encoding since it reduces both codebook generation time as well as closest codeword search time.

7. REFERENCES

[1] Ahmed Swilem. 2010. A fast vector quantization encoding algorithm based on projection pyramid with hadamard transformation. Image and Vision Comput., 28, (12), pp. 1637-1644.

[2] S. Annadurai, M. Sunderasen. 2009. Wavelet based color image compression relying on subband vector quantization, ICGST-GVIP J., 9, (1), pp.9-16.

[3] Baskar Ramamoorthy, Allen Gersho. 1986. Classified vector quantization of images, IEEE Trans. on Commun., 34, (11), pp.1105-1115.

[4] Chaur-Heh Hsieh, Wei-Yang Shao, Ming-Haw Jing. 2000. Image compression based on multi-stage vector quantization, J. of Visual Commun. and Image Represent., 11, (4), pp.374-384.

[5] Chun-Wei Tsai, Chao-Yang Lee. 2009. A fast VQ codebook generation algorithm via pattern reduction, Pattern Recognition Lett., 30,(7), pp.653-660.

[6] Gersho. A., Gray. R. M.. 1994. Vector quantization and signal compression, Kluwer Academic Publishers, 1994.

[7] Guobin Shen, Bing Zeng, Ming-L.Liou. 2003. Adaptive vector quantization with codebook updating based on locality and history, IEEE Trans. on Image Process., 12,(3), pp.283-295.

[8] Giuseppe Campobello, Mirko Mantineo, Giuseppe Patane, Marco Russo. 2005. LBGS: a smart approach for very large sets vector quantization, Signal Process: Image Commun., 20, (1), pp.91-114.

[9] Hsien-wen Tseng, ChinChu Chang. 2005. A very low bit rate image compression using transformed classified vector quantization, Informatica, 29, pp.335-341.

[10] Jim Z.C.Lai, Yi-Ching Liaw. 2009. A novel encoding algorithm for vector quantization using transformed codebook, Pattern Recognit. Lett., 42, (11), pp.3065-3070.

[11] Jim.Z.C.Lai, Yi-Ching Liaw, Julie Liu. 2008. A fast VQ codebook generation algorithm using codeword displacement, Pattern Recognit., 41,(1), pp.315-319.

[12] Matsumoto Hiroki, Kichikawa Fumito, Sasazaki Kazuya, Maeda Junji, Szuki Yukinori. 2010. Image compression using vector quantization with variable block size division, IEEJ Electronics, Information and Syst., 130, (8), pp.1431-1439.

[13] Nasser M. Nasrabad, Yushu.Feng. 1990. Image compression using address vector quantization, IEEE Trans. on Commun., 38, (12), pp.2166-2173.

[14] Paul Shelley, Xiaobo Li, Bin Han. 2004. A Hybrid quantization scheme for image compression, Image and Vision Comput., 22, (3), pp.203- 213.

[15] Robert Li, Jung Kim. 2000. Image compression using fast transformed vector quantization, IEEE Proc. of Appl. Imagery Pattern Recognit. Workshop, pp.141-145.

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

[17] Yoseph Linde, Andres Buzo, Robert M.Gray. 1980. An algorithm for vector quantizer Design, IEEE Trans. on Commun., 28, (1), pp.84-94.

[18] Yu-Chen Hu, Bing-Hwang Su, Chin-Chiang Tsou. 2008. Fast VQ codebook search algorithm for grayscale image coding, Image and Vision Comput., 26, (5), pp.657-666.