Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 51(65), Fascicola 2, 2006

Algorithms for Fast Full Nearest Neighbour Search on Unstructured Codebooks: A Comparative Study

Spiridon Florin Beldianu[1]

Abstract – This paper presents several fast nearest neighbor search algorithms for vector quantization on unstructured codebooks of arbitrary size and vector dimension that uses linear projections and variance of a vector. Several new inequalities based on orthonormal Tchebichef moments and projections on the first vectors of the DCT and PCA transformations of an image block are introduced to reject those codewords that are impossible to be the nearest codevector and cannot be rejected by inequalities based on Hadamard Transform, sum and variance, thereby saving a great deal of computational time, while introducing no extra distortion compared to the conventional full search algorithm.

Keywords: vector quantization, fast full nearest neighbor search, image vector quantization, linear projections

I. INTRODUCTION

Vector Quantization (VQ) [1], [2] is an efficient technique for data compression which has been successfully used in various applications involving VQ-based encoding and VQ-based recognition. The response time of encoding and recognition is a very important factor to be considered for real-time applications. The k-dimensional, N-level vector quantizer is defined as a mapping from a k-dimensional Euclidean space into a certain finite set $C=\{C_1,C_2,...,C_N\}$. The subset C is called a codebook and its elements are called codewords. The codeword searching problem in VQ is to assign one codeword to the input test vector thus the distortion between this codeword and the test vector is the smallest among all codewords. Given one codeword $C_j=\left(c_{j1},c_{j2},...,c_{jk}\right)$ and the test vector $\mathbf{x}=\left(x_1,x_2,...,x_k\right)$, the squared Euclidean distortion measure can be expressed as follows:

$$D(C_j,\mathbf{x})=\sum_{i=1}^{k}\left(c_{ji}-x_i\right)^2 . \qquad (1)$$

From the above equation, each distortion calculation requires multiplications and $2k-1$ additions. For an exhaustive full search algorithm, encoding each input vector requires N distortion computations and $N-1$ comparisons. Therefore, it is necessary to perform kN multiplications, $\left(2k-1\right)N$ additions and $N-1$ comparisons to encode each input vector. The need for a larger codebook size and higher dimension for high performance in VQ encoding system results in increased computation load during the codeword search.

Many researchers have looked for fast encoding algorithms to accelerate the VQ process. These works can be classified into two groups. The first group rely on the use of data structures that facilitate fast search of the codebook such as TSVQ or K-d tree [3] ,[4]. The second group addresses an exact solution of the nearest-neighbor encoding problem on unstructured codebooks. A very simple but effective method is the partial distortion search (PDS) method reported by Bei and Gray [5], which allows early termination of the distortion calculation between a test vector and a codeword by introducing a premature exit condition in the searching process. The equal-average nearest neighbor search (ENNS) algorithm uses the mean value of an input vector to reject impossible codewords [6]. The improved algorithm, i.e., the equal-average equal-variance nearest neighbor search (EENNS) algorithm, uses the variance as well as the mean value of an input vector to reject more codewords [7]. This algorithm reduces computational time further with $2N$ additional memory cells. The improved algorithm termed IEENNS uses the mean and the variance of an input vector like EENNS but develops a new inequality between these features and the distance [8],[9]. The DHSS3 [10] method uses an inequality based on projections on the firsts three axis of ordered Walsh-Hadamard transformation to reject impossible codewords. In [11] is presented a new algorithm based on projections on Tchebichef Moments (also named as Discreete Tchebichef Transform—DTT) vector basis (DTTS), which proves to have a lower search complexity than IEENNS.

In this paper, we will examine the kernel and the complexity search for IEENNS, DHSS3, DTTS algorithms and two new ones based on projections on

[1] Faculty of Electronics and Telecommunications,Telecommunications Department,
Bd. Carol 11, 700506, Iasi, fbeldianu@etc.tuiasi.ro

BUPT

three vector basis of DCT and Karhunen Loeve (KLT) transformations.

II. THE ALGORITHM

A. *IEENNS and DHSS3 algorithms*

The IEENNS algorithm [8] uses two characteristics of a vector, sum and the variance simultaneously. Let $\mathbf{x} = [x_1, x_2, ..., x_k]$ be a k-dimensional vector. The sum of vector components can be expressed as $S_{\mathbf{x}} = \sum_{i=1}^{k} x_i$ and the variance as $V_{\mathbf{x}} = \sqrt{\sum_{i=1}^{k} (x_i - S_{\mathbf{x}} / k)}$. The basic inequalities for IEENNS method are as follows: if $\mathbf{y}$ is a codeword and $\mathbf{x}$ is an input vector, the following important inequalities are true:

$$\begin{aligned} &\left(S_{\mathbf{x}} - S_{\mathbf{y}}\right)^2 \le kD(\mathbf{x}, \mathbf{y}) \\ &\left(S_{\mathbf{x}} - S_{\mathbf{y}}\right)^2 + k\left(V_{\mathbf{x}} - V_{\mathbf{y}}\right)^2 \le kD(\mathbf{x}, \mathbf{y}) \end{aligned} \tag{2}$$

Assuming that the current minimum distortion is $D_{\min}$, the main spirit of the IEENNS algorithm can be stated as follows:
If $(S_{\mathbf{x}} - S_{C_j})^2 \ge kD_{\min}$ **then** $D(\mathbf{x}, C_j) \ge D_{\min}$ and C_j will not be the nearest neighbor to $\mathbf{x}$; **ElseIf** $(V_{\mathbf{x}} - V_{C_j})^2 \ge D_{\min}$ **then** $D(\mathbf{x}, C_j) \ge D_{\min}$ and C_j will be rejected; **ElseIf** $(S_{\mathbf{x}} - S_{C_j})^2 + k(V_{\mathbf{x}} - V_{C_j})^2 \ge kD_{\min}$ **then** $D(\mathbf{x}, C_j) \ge D_{\min}$ and C_j will be rejected; **Else** compute $D(\mathbf{x}, C_j)$ and if $D(\mathbf{x}, C_j) < D_{\min}$ update $D_{\min} = D(\mathbf{x}, C_j)$. To perform the IEENNS algorithm, $2N$ values should be computed off-line and stored.
The DHSS3 algorithm [10] utilizes the compactness property of signal energy on transform domain and the geometrical relations between the input vector and every codevector to eliminate those codevectors that have no chance to be the closest codeword of the input vector. It achieves a full search equivalent performance. Let $\mathbf{h}_1$, $\mathbf{h}_2$ and $\mathbf{h}_3$ be the first three orthonormal vectors of ordered Walsh-Hadamard transform. For example if $k = 16$, we have:

$$\begin{aligned} \mathbf{h}_1 &= [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]/4; \\ \mathbf{h}_2 &= [1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1]/4; \\ \mathbf{h}_3 &= [1,1,-1,-1,1,1,-1,-1,1,1,-1,-1,1,1,-1,-1]/4; \end{aligned}$$

Denote the axis in the direction of $\mathbf{h}_i (i = 1,2,3)$ as the i-th axis. Let $H_i(\mathbf{x})$ be the projection value of an input vector $\mathbf{x}$ on the i-th axis. That is, $H_i(\mathbf{x})$ is the inner product of $\mathbf{x}$ and $\mathbf{h}_i$, and can be calculated as follows: $H_i(\mathbf{x}) = \langle \mathbf{x}, \mathbf{h}_i \rangle$.
It can been shown that for an input vector $\mathbf{x}$ and for a codeword C_j the following inequality is true:

$$D(\mathbf{x}, C_j) \ge \sum_{i=1}^{3} \left| H_i(\mathbf{x}) - H_i(C_j) \right|^2 \tag{3}$$

To speed up the searching process, all codewords are sorted in ascending order of their projections on the first axis. The elimination process of the DHSS3 algorithm consists of four steps. The firsts three steps are as follows:
If $\left| H_i(\mathbf{x}) - H_i(C_j) \right| \ge \sqrt{D_{\min}}$ $(i = 1,2,3)$ then C_j will be rejected. Last step is: **If** $\sum_{i=1}^{3} \left| H_i(\mathbf{x}) - H_i(C_j) \right|^2 \ge \sqrt{D_{\min}}$ then C_j will be rejected; **ElseIf** $D(\mathbf{x}, C_j) < D_{\min}$ update $D_{\min} = D(\mathbf{x}, C_j)$. To perform the DHSS3 algorithm, $3N$ values should be computed off-line and stored.

B. *Tchebichef Polynomials and Orthonormal Moments*

For a given positive integer (usually the image size), and a value x in the range $[0, M-1]$, the scaled Tchebichef polynomials $t_n(x)$, $n = 0.1, ..., M-1$, are defined using the following recurrence:

$$t_n(x) = \frac{(2n-1)t_1(x)t_{n-1}(x) - (n-1)\left(1 - \frac{(n-1)^2}{M^2}\right)t_{n-2}(x)}{n} \qquad n = 2,3,...M-1 \tag{4}$$

where $t_0(x) = 1$ and $t_1(x) = (2x + 1 - M)/M$. The above definition uses the following scaled factor [12] for the polynomial of degree n:

$$\beta(n, M) = M^n \tag{5}$$

The set $\{t_n\}$ has a squared-norm given by:

$$\begin{aligned} \rho(n, M) &= \sum_{x=0}^{M-1} \{t_n(x)\}^2 = \\ &= \frac{M(1 - 1/M^2)(1 - 2^2/M^2)...(1 - n^2/M^2)}{2n+1} \end{aligned} \tag{6}$$

These polynomials are orthogonal, and by modifying the scale factor $\beta(n, M)$ in (5) as in [13]:

$$\beta(n, M) = \sqrt{\frac{M(M^2-1)(M^2-2^2)...(M^2-n^2)}{2n+1}} \tag{7}$$

we obtain a set of orthonormal polynomials that can be used to define a set of orthonormal moments in (8).

$$T_{m,n}(f) = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \hat{t}_m(x) \hat{t}_n(x) f(x, y) \qquad m, n = 0,1,...M-1 \tag{8}$$

$f(x, y)$ denotes the intensity value of the pixel position (x, y) in the image. It can be easily seen that the recurrence relations given in (4) now change to the following:

$$\hat{t}_n(x) = \alpha_1 x \hat{t}_{n-1}(x) + \alpha_2 \hat{t}_{n-1}(x) + \alpha_3 \hat{t}_{n-2}(x) \qquad n = 2,3,...M-1; \quad x = 0,1,2,...M-1 \tag{9}$$

BUPT

where:

$$\alpha_1 = \frac{2}{n}\sqrt{\frac{4n^2-1}{M^2-n^2}},\ \alpha_2 = \frac{(1-M)}{n}\sqrt{\frac{4n^2-1}{M^2-n^2}},$$

$$\alpha_3 = \frac{(n-1)}{n}\sqrt{\frac{2n+1}{2n-3}}\sqrt{\frac{M^2-(n-1)^2}{M^2-n^2}}. \quad (10)$$

The starting values for the above recursion can be obtained from the following equations:

$$\hat{t}_0(x) = \frac{1}{\sqrt{M}}$$
$$\hat{t}_1(x) = (2x+1-M)\sqrt{\frac{3}{M(M^2-1)}}. \quad (11)$$

The squared norm is now $\rho(n,M) = \sum_{i=0}^{M-1}\{\hat{t}_n(i)\}^2 = 1$.

Since the new moment set is orthonormal we can introduce the following theorem which is an inequality between Euclidian distance of two images and sum of squared differences of orthonormal Tchebichef moments of those images.

Theorem: Let f and g be two images with $M\times M$ resolution. Then:

$$\sum_{m=0}^{p\le M-1}\sum_{n=0}^{q\le M-1}\left|T_{mn}(f)-T_{mn}(g)\right|^2 \le D(f,g) \quad (12)$$

where $D(f,g)$ is the squared Euclidean distance between images f and g, and can be defined similar as in (1).

Proof: Since $m,n = 0,1,2...M-1$, the set $\{T_{mn}\}$ is composed by M^2 orthonormal moments. So, $T_{mn}(f)$ can be assimilated with a linear orthonormal transformation of an image f which has M^2 vector basis. A linear orthonormal transformation is a bijective map between two metric spaces which preserves the distances. This property is called isometry, and in this case we can write:

$$\sum_{m=0}^{M-1}\sum_{n=0}^{M-1}\left[T_{mn}(f)-T_{mn}(g)\right]^2 = D(f,g). \quad (13)$$

The left side of (13) is the squared Euclidean distance computed in the output space of the transformation given by Tchebichef moments. Having this equality is obviously that the inequality in (12) always holds.

For example in Fig. 1 are presented the firsts four vector basis of this linear transform for $M=4$. $T_{00}(f), T_{01}(f), T_{10}(f), T_{11}(f)$ can be computed using the dot product between those vector basis and input image f.

C. DTTS Algorithm

For the proposed algorithm we use only firsts three moments [3], namely, T_{pq}, where $(p,q)\in\{(0,0),(0,1),(1,0)\}$. The inequality in (12) becomes now:

$$\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix} \quad \frac{1}{\sqrt{80}}\begin{bmatrix}-3&-1&1&3\\-3&-1&1&3\\-3&-1&1&3\\-3&-1&1&3\end{bmatrix}$$

$$\frac{1}{\sqrt{80}}\begin{bmatrix}-3&-3&-3&-3\\-1&-1&-1&-1\\1&1&1&1\\3&3&3&3\end{bmatrix} \quad \frac{1}{20}\begin{bmatrix}9&3&-3&-9\\3&1&-1&-3\\-3&-1&1&3\\-9&-3&3&9\end{bmatrix}$$

Figure 1. From left to right and from top to bottom: firsts four vector basis used to compute orthonormal Tchebichef moments T_{00}, T_{01}, T_{10} and T_{11}.

$$(T_{00}(f)-T_{00}(g))^2 + (T_{01}(f)-T_{01}(g))^2 + $$
$$+(T_{10}(f)-T_{10}(g))^2 \le D(f,g) \quad (14)$$

The proposed searching sequence for a given input image $f\in\mathbf{F}\equiv\{f_1,f_2,\ldots,f_L\}$ can be described as follows:

*

Step 0: For every image codeword C_j, $j=\overline{1,N}$, $T_{00}(C_j), T_{01}(C_j), T_{10}(C_j)$ are computed. The codewords are sorted in the ascending order of $T_{00}(C_j)$. This step is operated off-line. In the following steps the memory for $T_{00}(C_j), T_{01}(C_j), T_{10}(C_j)$ $j=1,2,...,N$ are ready; Go to step **1**;

For every input image vector $f\in\mathbf{F}$ find the nearest neighbor codevector as follows:

Step 1: $T_{00}(f), T_{01}(f), T_{10}(f)$ are computed; go to step **2**

Step 2: Obtain the tentative matching codeword C_p whose index is calculated by $p=\arg\min_j\left|T_{00}(f)-T_{00}(C_j)\right|$. Calculate the squared Euclidian distortion $D_{\min} = D(f,C_p)$ and set $i=1$; go to step **3**;

Step 3: ***If*** $p+i>N$ or codeword C_{p+i} to C_N have been rejected go to step **4**; ***Else*** go to step **3.1**;

Step 3.1: ***If*** $\left|T_{00}(f)-T_{00}(C_{p+i})\right| \ge \sqrt{D_{\min}}$ reject the codewords C_{p+i} to C_N and go to step **4**; ***Else*** go to step **3.2**;

Step 3.2: ***If*** $\left|T_{01}(f)-T_{01}(C_{p+i})\right| \ge \sqrt{D_{\min}}$ reject the codeword C_{p+i} and go to step **4**; ***Else*** go to step **3.3**;

Step 3.3: ***If*** $\left|T_{10}(f)-T_{01}(C_{p+i})\right| \ge \sqrt{D_{\min}}$ reject the codeword C_{p+i} and go to step **4**; ***Else*** go to step **3.4**;

Step 3.4: ***If***

$$\left|T_{00}(f)-T_{00}(C_{p+i})\right|^2 + \left|T_{01}(f)-T_{01}(C_{p+i})\right|^2 +$$
$$+\left|T_{10}(f)-T_{10}(C_{p+i})\right|^2 \ge D_{\min}$$

BUPT

reject the codeword C_{p+i} and go to step 4; **Else** use PDS to find minimum distortion, update $D_{\min} = \min(D_{\min}, D(f, C_{p+i}))$ and go to step **4**;

Step 4: **If** $p-i<1$ or codeword C_{p-i} to C_1 have been rejected go to step **5**; **Else** go to step **4.1**

Step 4.1: **If** $\left|T_{00}(f) - T_{00}(C_{p-i})\right| \geq \sqrt{D_{\min}}$ reject the codewords C_{p-i} to C_1 and go to step **5**; **Else** go to step **4.2**;

Step 4.2: **If** $\left|T_{01}(f) - T_{01}(C_{p-i})\right| \geq \sqrt{D_{\min}}$ reject the codeword C_{p-i} and go to step **5**; **Else** go to step **4.3**;

Step 4.3: **If** $\left|T_{10}(f) - T_{01}(C_{p-i})\right| \geq \sqrt{D_{\min}}$ reject the codeword C_{p-i} and go to step **5**; **Else** go to step **4.4**;

Step 4.4: **If**

$$\left|T_{00}(f) - T_{00}(C_{p-i})\right|^2 + \left|T_{01}(f) - T_{01}(C_{p-i})\right|^2 + \\ + \left|T_{10}(f) - T_{10}(C_{p-i})\right|^2 \geq D_{\min}$$

reject the codeword C_{p-i} and go to step **5**; **Else** use PDS to find minimum distortion, update $D_{\min} = \min(D_{\min}, D(f, C_{p-i}))$ and go to step **5**;

Step 5: Set $i = i+1$; **If** $p+i>N$ and $p-i<1$ or all codewords have been deleted, terminate the algorithm and return the closest codeword for input image vector f; **Else** go to step **3**.

*

The complexity reduction is caused to reduction in number of addition and multiplications needed to compute the left side of (11) instead to compute $D(f, C_i)$ in (1). By choosing this searching sequence, experimental results shows that this proposed algorithm is faster than IEENNS and DHSS3 algorithms, in terms of computational complexity.

D. *DCT and PCA based algorithms*

Similar as in C section, we can develop two new algorithms which uses instead of several projections on DTT, three projections on firsts DCT or PCA vector basis. We have to note that in PCA based approach we must previously compute the first three eigen vectors corresponding to the greatest eigenvalues of the covariance matrix of the codebook. Being the fact that the DCT and PCA are orthonormal transformations, the *Theorem* is also true for this two approaches. The new methods are the same as DTT-S except that we replace in (12), (13) and (14) the Tchebichef moments $T_{mn}(f)$ with the projections on the first three basis vectors of the DCT and PCA transformation Also note that in some practical applications additional computation of the eigen vectors for PCA based method, and for some codebooks, can be sometimes prohibitive.

Table 1. Comparison of average Number of Distortion Calculations per Image (4×4) Block

Codebook size	Method	Encoded image	
		Peppers	**Baboon**
128	Full Search	128	128
	PDS	55.65	89.32
	DHSS3	3.97	16.84
	IEENNS	3.59	14.96
	DTTS	**2.34**	**11.85**
	DCT based	**2.32**	**12.01**
	PCA based	**2.28**	**11.43**
512	Full Search	512	512
	PDS	174.34	302.23
	DHSS3	13.09	64.16
	IEENNS	12.30	53.97
	DTTS	**7.01**	**46.17**
	DCT based	**6.96**	**46.23**
	PCA based	**6.81**	**43.82**
1024	Full Search	1024	1024
	PDS	486.23	743.21
	DHSS3	24.65	114.60
	IEENNS	22.95	89.66
	DTTS	**12.92**	**82.01**
	DCT based	**12.85**	**82.08**
	PCA based	**12.08**	**79.30**

III. EXPERIMENTAL RESULTS

The images used in this experiment are 512×512 monochrome with 256 gray levels. An image is partitioned in 4×4 image blocks and the codebook is design using the Linde-Buzo-Gray (LBG) algorithm with Lena image as a training set. The Peppers and Baboon images are used as the test images. The proposed algorithms are compared to the Full Search, PDS [5] , IEENNS [8,9] and DHSS3 [10] algorithms. Table I and II show the average number of distortion computations and the number of operations (multiplications, additions and comparisons) per pixel for various codebook sizes. For the DCT based algorithm the projections are choosen as the first three elements from the matrix of the DCT-2D coefficients, namely 00, 01 and 10, and for PCA based method are choosen the eigenvectors corresponding with the first three eigenvalues in decreasing order.

From Table I, we can see that our methods have the best performance of rejecting unlikely codewords. Compared with IEENNS method, proposed algorithms can reduce the number of distortion calculations by 10% to 44% and the average reduction of operations per pixel needed to encode an image block is 39% for Peppers and 11% for Baboon. Compared with DHSS3, our approaches also reduce the number of distortion calculations by 13% to 50% and the average reduction of operations is 43% for Peppers and 15% for Baboon. Compared with DHSS3 and IEENNS, DTT, DCT-2D and PCA based methods can extract much better the information about spatial orientation of image blocks in k-dimensional space. So, they can better discriminate between images with different features, which will determine an increased number of rejected codewords.

Also note that: **(i)** The complexity search for Peppers is approximatively 20%, 17% and 16% for 128, 512

BUPT

Table 2. Comparison of average Number of Operations per Pixel

Codebook size	Search Method	Encoded image					
		Peppers			**Baboon**		
		Mult.	Add.	Comp.	Mult.	Add.	Comp.
128	Full Search	128	248	8	128	248	8
	PDS	19.44	52.16	4.67	40.66	96.48	8.52
	DHSS3	5.01	9.99	2.1975	20.9462	40.3312	7.3831
	IEENNS	5.40	11.34	1.73	19.35	36.64	5.72
	DTTS	**2.98**	**6.54**	**1.94**	**15.08**	**30.05**	**6.74**
	DCT based	**2.972**	**6.52**	**1.938**	**15.16**	**31.03**	**6.88**
	PCA based	**2.91**	**6.48**	**1.83**	**14.84**	**29.59**	**6.55**
512	Full Search	512	992	32	512	992	32
	PDS	57.60	147.23	16.28	143.38	339.71	32.98
	DHSS3	16.64	33.42	7.69	79.80	153.47	27.94
	IEENNS	16.05	32.01	6.07	67.12	125.19	21.13
	DTTS	**9.11**	**20.58**	**6.73**	**58.75**	**116.37**	**25.62**
	DCT based	**9.08**	**20.32**	**6.69**	**58.92**	**116.97**	**25.60**
	PCA based	**8.97**	**19.92**	**6.67**	**57.01**	**110.99**	**24.10**
1024	Full Search	1024	1984	64	1024	1984	64
	PDS	104.45	262.21	29.77	263.87	574.32	59.52
	DHSS3	31.49	63.06	14.55	142.69	274.31	49.99
	IEENNS	29.10	57.28	11.36	111.06	207.81	36.81
	DTTS	**16.91**	**38.25**	**12.68**	**98.56**	**197.81**	**39.87**
	DCT based	**16.38**	**38.17**	**12.69**	**99.87**	**198.23**	**39.90**
	PCA based	**16.32**	**37.89**	**11.93**	**94.76**	**189.8**	**37.06**

and 1024 codebook size from the complexity search of the Baboon; **(ii)** PCA based approach seems to be slightly better than DTT and DCT based approaches especially for large codebooks. This is an expected result because PCA is the optimal transform regarding the compaction of the energy. But if we consider the fact that we have to use supplementary computation to obtain the eigen vectors, the performance of the overall PCA based method may have a drawback; **(iii)** The complexity difference between DTT and DCT based algorithms is reduced. An explanation is that the kernels of the DCT and DTT transformation are both derived from orthogonal Tchebichef polynomials. From table I and II we observe that for Peppers image, DCT based method outperforms the DTTS and for Baboon is the opposite case; **(iv)** The average time needed for encoding a specific image also depends on two factors: how complex is the image, which refers to how larger is the entropy of that image (it is clear that Baboon has larger entropy than Peppers) and the machine which implements the encoding algorithm. There are several machines in which a multiplication requires much more time than an addition or a comparison, and are others where the difference is not so significant. Also, the floating or integer point implementation can cause reordering of the performance of the presented methods; **(v)** At last but not at least, the accessing time for the precomputed values can be different on several types of implementations.

In conclusion, the trade-off between those factors may produce a system which spent significant less time than in the exhaustive search.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, some fast-encoding algorithms are presented and new ones are introduced. We have presented a new inequality between Euclidian distance of two image blocks and sum of squared differences of orthonormal Tchebichef Moments (also first three projections on the DCT and KLT transforms). This algorithm uses projections of an image block to eliminate many of the unlikely codewords, which cannot be rejected by other available algorithms. Compared with other available approaches, our algorithm has the best performance in terms of number of distortion calculations and the number of operations per pixel needed to encode a certain image.

Future work will focus on using image blocks with 8×8 resolution and will include higher order Tchebichef Moments in (14), which will reject more codewords that cannot be rejected by presented methods.

REFERENCES

[1] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design", *IEEE Trans. Commun.*, vol COM-28, no. 1, pp.84-95, 1980

[2] A. Gersho and R. M. Gray, *Vector quantization and signal compression*, Kluwer Academic Press, Massachusetts, 1990

[3] N. Moayeri, D. L. Neuhoff, and W. E. Stark, "Fine-coarse vectorquantization," *IEEE Trans. Signal Processing*, vol. 39, pp. 1503–1515, July 1991.

[4] V. Ramasubramanian and K. K. Paliwal, "Fast k-dimensional tree algorithms for nearest neighbor search with application to vector quantization encoding," *IEEE Trans. Signal Processing*, vol. 40, pp. 518–531, Mar. 1992

[5] C. D. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithms for vector quantization and pattern matching," *IEEE Trans. Commun.*, vol. COMM-33, pp. 1132–1133, Oct. 1985.

[6] S. W. Ra and J. K. Kim, "A Fast Mean-Distance-Ordered Partial Codebook Search Algorithm for Image Vector Quantization," *IEEE Trans. Circuits Syst. II*, vol. 40, no. 9, pp. 576–579, 1993

BUPT

[7] C. H. Lee and L. H. Chen, "Fast closest codeword search algorithm for vector quantization," *Proc. Inst. Elect. Eng.*, vol. 141, no. 3, pp. 143–148, 1994.

[8] S. J. Baek, B. K. Jeon, and K. M. Sung, "A fast encoding algorithm for vector quantization," *IEEE Signal Processing Lett.*, vol. 4, pp. 325–327, Feb. 1997.

[9] J.-S. Pan, Z.-M. L. Lu, and S.-H. Sun, "An Efficient Encoding Algorithm for Vector Quantization Based on Subvector Technique", *IEEE Trans. Image Processing*, vol. 12, no. 3, March 2003

[10] S.C. Tai, C.C. Lai, and Y.C. Lin, "Two fast nearest neighbor searching algorithms for image vector quantization", *IEEE Trans. Commun*., vol. 40, pp.1623-1628, Dec.1996

[11] S.F. Beldianu, "A Fast Vector Quantization image encoding using Tchebichef Moments", *Proceedings of the 4-th International Conference on "Microelectronics and Computer Science"*, vol. 2, september 2005, pp.40-43

[12] R. Mukundan, S.H. Ong, P.A. Lee, "Image Analysis by Tchebichef Moments", *IEEE Transaction on Image Processing*, vol. 10, No. 9, pp.1357-1364, September 2001

[13] R. Mukundan, "Some Computational Aspects of Discrete Orthonormal Moments", *IEEE Trans. on Image Processing*, vol. 13, No. 8, August 2004