A Novel Fast Zigzag Prune
4×4 Discrete Tchebichef Moment Based Image Compression Algorithm

Ranjan K. Senapati¹, Umesh C. Pati², Kamala K. Mahapatra³
Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela
Orissa, India.

¹ranjankumarsenapati@gmail.com, ²ucpati@nitrkl.ac.in, ³kkm@nitrkl.ac.in

Abstract— The Discrete Tchebichef Moment (DTM) is a linear orthogonal transform which has higher energy compactness property like other orthogonal transforms. It is recently found applications in image analysis and compression. This paper proposes a new approach of fast zigzag pruning algorithm of 4×4 DTM coefficients. The principal ideal of the proposed algorithm is to make use of the distributed arithmetic and the symmetry property of 2-D DTM, which combines the similar terms of the pruned output. The multiplication terms are replaced by shift and add operations so as to reduce the computation. Equal number of zigzag pruned coefficients and block pruned coefficients are used for comparison to test the efficiency of our algorithm. Experimental method shows that our method is competitive with the block pruned method. Specifically for 3×3 block pruned case our method provides lesser computational complexity and has higher peak signal to noise ratio (PSNR).

Keywords— Discrete Tchebichef Moment, Image compression, Fast algorithm, Block Prune, Zigzag Prune.

I. INTRODUCTION

Image Transform methods using orthogonal kernel functions are commonly used in image compression. One of the most widely known image transform method is Discrete Cosine Transform (DCT), which is used in JPEG compression standard [1]. Battery operated smaller computing devices such as Personal digital assistants (PDAs), digital cameras, mobile phones require a lot of extensive image processing. Usually DCT based compression techniques are used for these portable devices. However, DCT compression techniques need floating point operations, which consumes most of the processing time. Later, the development of integer cosine transform (ICT) which replaces DCT, reduces the computation to a much greater extent because ICT requires evaluation of integer operations instead of floating point operations [2]. The discrete Tchebichef moment (DTM) compression first proposed by Mukundan [3] could be an alternative to ICT for these portable devices, because, DTM needs only evaluation of integer operations for transform. Furthermore, it has additional properties such as:

(i) Its discrete representation exactly matches with image coordinates.
(ii) Absence of numerical approximation, gives rise to more accurate image feature representation [4].

The DTM transform can not only applied to continuous images but also for generic artificial images generated by graphical software.

There are many DCT compression algorithms which can be computed in a fast way by means of direct or indirect methods. A direct polynomial transform technique for two dimensional (2-D) DCT is proposed by Duhamel et al. [5]. Feig and Winograd [6] proposed another fast algorithm for direct cosine transform. Vetterli [7] proposed an indirect method to calculate 2-D DCT by mapping it into a 2-D DFT plus a number of rotations.

The above algorithms assume same number of input and output points. However, in image coding applications, the most useful information about the image data is kept in the low-frequency DCT coefficients. Therefore, only these coefficients could be computed. This gives rise to the application of pruning technique. Using this idea, additional processing speed is also possible.

Several algorithms for pruning the 1-D DCT in [8]-[11] and 2-D DCT in [12]-[14] has been addressed. Ref. [8], has shown that output-pruned DCT and DST can be computed by slightly modifying output-pruned FFT algorithms for real valued data of the same size. A recursive pruning DCT algorithm has been presented in [9] with a structure that allows the generation of next higher order pruned DCT from two identical lower order pruned DCTs. A fast pruning algorithm proposed in [10] computes N₀ lowest frequency components of length-N discrete cosine transform, where N₀ is any integer. A generalized output pruning algorithm for matrix–vector multiplications is proposed in [11], which eliminate thoroughly the unnecessary operations for computing an output pruning DCT. Peng [12] has presented a DCT-based computational complexity scalable video decoder via properly pruning the DCT data. Experimental results showed that the complexity can be scaled from 100% to 38% with graceful quality degradation. Walmsley et al. [13] uses pruning method in JPEG standard. They have shown that for an 8×8 image block it is only necessary to calculate a 4×4 subset of DCT values to retain acceptable image quality. The effects of pruning on parallelization and speedup process are also discussed. The pruning algorithm in [13] and [14] computes a set of coefficients included in a top-left triangle. It-
corresponds to zigzag scanning where all coefficients in each diagonal are computed. In [8] and [12], a sub-block of coefficients out of \( N \times N \) is computed.

The discrete Tchebichef moment (DTM) transform is another linear orthonormal version of orthogonal Tchebichef polynomials, that has very similar energy compactness for natural and artificial images. Recently, 4\( \times 4 \) DTM fast algorithms for image compression have been proposed [15]-[18]. A 2\( \times 2 \) block pruned out of 4\( \times 4 \) DTM algorithm which computes the upper left quarter of 4\( \times 4 \) image blocks is proposed [17]. Saleh [18] proposed a fast 4\( \times 4 \) algorithm suitable for different block sizes.

In this paper we propose a fast zigzag pruning DTM algorithm of different prune lengths. A comparison with the existing DTM fast algorithms available in the literature till date is made. Finally a comparison is made between zigzag pruning DTM reconstructed image quality and corresponding block pruned reconstructed image quality.

The organization of the paper is as follows: Section II describes the mathematical definition of Discrete Tchebichef Moment Algorithm. Some properties of DTM transform is presented in Section III. In Section IV a fast forward zigzag pruning discrete Tchebichef moment (DTM) algorithm is proposed. Section V discusses the computational complexity of the proposed method. Simulation results and analysis is carried out in Section VI. Finally we conclude in Section VII.

II. DISCRETE TCHEBICHEF TRANSFORM

The Discrete Tchebichef Moment (DTM) transform uses Tchebichef moments to provide a basis matrix. As with DCT, the DTM is derived from the orthonormal Tchebichef polynomials. DTM exhibits similar energy compactation properties like DCT [4]. For image of size \( N \times N \), the forward Discrete Tchebichef Transform of order \( p+q \) is defined as:

\[
T_{pq} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} t_p(m) t_q(n) f(m,n) \quad \text{(1)}
\]

The inverse transformation of DTM is defined by:

\[
f(m,n) = \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} T_{pq} t_p(m) t_q(n) \quad \text{(2)}
\]

Equation (4) can also be expressed using a series representation involving matrices as follows:

\[
f(i,j) = \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} T_{pq} \tau_{pq}(i,j) \quad \text{(3)}
\]

where \( \tau_{pq} \) is an 8\( \times \)8 matrix (called basis images) and is defined as:

\[
\tau_{pq} = \begin{bmatrix}
    t_p(0) t_q(0) & t_p(0) t_q(1) & \cdots & t_p(0) t_q(7) \\
    t_p(1) t_q(0) & t_p(1) t_q(1) & \cdots & t_p(1) t_q(7) \\
    \vdots & \vdots & \ddots & \vdots \\
    t_p(7) t_q(0) & t_p(7) t_q(1) & \cdots & t_p(7) t_q(7)
\end{bmatrix}.
\quad \text{(4)}
\]

The 2-dimensional (2-D) basis function of the DTM is defined as follows:

\[
\tau(p,m,q,n) = t_p(m) t_q(n). \quad \text{(5)}
\]

where, \( t_p(m) \) and \( t_q(n) \) are \( p^{th} \) and \( q^{th} \) order Tchebichef-moments respectively. These can be defined using the following function over the discrete range \( [0, N] \).

\[
t_p(x) = p! \sum_{k=0}^{p} (-1)^k \binom{N-1-k}{p-k} \binom{p+k}{k}. \quad \text{(6)}
\]

Due to the large dynamic range of the intermediate values generated by (6), it is not feasible to calculate the values of DTM on a point wise basis. Instead we calculate the function using the following recurrence relation:

\[
t_0(m) = \frac{1}{\sqrt{N}}, \quad \text{(7)}
\]

\[
t_1(m) = (2m+1-N) \frac{3}{N(N^2-1)}, \quad \text{(8)}
\]

\[
t_{p+1} = (A_1 t_{p+1} + A_2 t_{p+2}) + A_3 t_{p-2} \quad \text{(9)}
\]

Where \( p \in [2..N] \), and coefficients \( A_1, A_2 \) and \( A_3 \) are as defined as follows:

\[
A_1 = \frac{2}{p} \sqrt{\frac{4p^2-1}{N^2-p^2}}, \quad \text{(10)}
\]

\[
A_2 = \frac{1-N}{p} \sqrt{\frac{4p^2-1}{N^2-p^2}},
\]

and \( A_3 = \frac{p-1}{p} \sqrt{\frac{2p+1}{N^2-(p-1)^2}} \) .

III. PROPERTIES OF DISCRETE TCHEBICHEF MOMENT

A. Separability

The definition of DTM can be written in separable form as:
\[ T_{pq} = \sum_{m=0}^{N-1} t_p(m) \sum_{n=0}^{N-1} t_q(n)f(m,n). \]  

(11)

Therefore, it can be evaluated using two dimensional transforms as follows:

\[ g_q(x) = \sum_{n=0}^{N-1} t_q(y)f(m,n), \]

(12)

\[ T_{pq} = \sum_{n=0}^{N-1} t_p(x)g_q(x). \]

(13)

From (12) and (13), it is clear that 2-D DTM and 2-D DCT are just one dimensional DTM and DCT applied twice by successive 1-D operations, once in x-direction, and the once in y-direction.

**B. Even Symmetry**

From [4], it can be shown that Tchebichef polynomials satisfy the property

\[ t_p(N-1-m) = (-1)^p t_p(m), \quad p = 0, 1, ..., N-1. \]  

(14)

The above two properties are commonly used in transform coding methods to get substantial reduction in the number of arithmetic operations.

**C. Orthogonality**

Fig. 1 shows the 2-D basis images of the DTM. In the basis images, it has been observed that the low frequencies reside in the upper left corner of the spectrum, while the high frequencies are in the lower right. The basis functions for rows are increasing frequencies in horizontal directions while the basis functions for columns are increasing frequencies in vertical directions.

**D. Energy Compaction**

DTM exhibits excellent energy compaction properties for highly correlated images. The energy of the image is packed into low frequency region (i.e. top left region) as shown in Fig. 2. As noted by Mukundan [3] the recurrence relation form (7)-(9), causes minor numerical errors to propagate through calculation. This error manifests itself in the collapse of basis function as shown in Fig. 2. By performing transform of image blocks of size less than 64×64 pixels this problem can be safely avoided.

**E. Tchebichef Polynomials**

Fig. 3 shows that the classical Tchebichef polynomials do not show large variation in dynamic range values. Therefore, numerical overflow errors do not occur for large images [3].

**IV. PROPOSED ZIGZAG PRUNED 4×4 DISCRETE TCHEBICHEF MOMENT ALGORITHM**

The 2-D Discrete Tchebif Moment transform from (1) can be expressed in matrix form as:

\[ T = \tau F \tau^T. \]  

(15)
where, \( F \) is the 2-D input data, \( \tau \) is the Tchebichef basis and \( T \) is the 2-D matrix of transformed coefficients. The transform kernel for 4 point DTM can be defined from (4) as:

\[
\tau = \begin{bmatrix}
1/2 & 1/2 & 1/2 & 1/2 \\
-3/2\sqrt{5} & -1/2\sqrt{5} & 1/2\sqrt{5} & 3/2\sqrt{5} \\
1/2 & -1/2 & -1/2 & 1/2 \\
-1/2\sqrt{5} & 3/2\sqrt{5} & -3/2\sqrt{5} & 1/2\sqrt{5}
\end{bmatrix}.
\] (16)

By defining \( x = 1/2 \) and \( y = 1/\sqrt{5} \), (16) can be written as:

\[
\tau = \begin{bmatrix}
x & x & x & x \\
-3xy & -xy & xy & 3xy \\
x & -x & -x & x \\
-xy & 3xy & -3xy & xy
\end{bmatrix}.
\] (17)

Factorizing (17) we will get

\[
S = \begin{bmatrix}
x & x & x & x \\
xy & xy & xy & xy \\
x & x & xy & xy \\
xy & xy & xy & xy
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
-3 & -1 & 1 & 3 \\
1 & -1 & -1 & 1 \\
-1 & 3 & 3 & 1
\end{bmatrix}.
\] (18)

\( S \) is a scaling matrix and can be separated from the core transform computation.

The expression in (15) can be factorized as:

\[
T = (\hat{\tau} F \hat{\tau}) \otimes \hat{S},
\] (19)

where

\[
\hat{S} = \begin{bmatrix}
x^2 & x^2 & x^2 & x^2 \\
x^2y & x^2y^2 & x^2y & x^2y^2 \\
x^2y & x^2y^2 & x^2y & x^2y^2 \\
x^2y & x^2y^2 & x^2y & x^2y^2
\end{bmatrix}
\]

Symbol \( \otimes \) indicates element-by-element multiplication. Since \( \hat{\tau} \) is orthogonal, but not orthonormal. Normalization can be done by merging \( \hat{S} \) into the quantization matrix.

By substituting (17) in (15), we can calculate each transformed coefficients for the input matrix \( F \). Furthermore, the even symmetry property allow us to group terms of the form \( f(x,1) \pm f(x,3) \) for \( x = 0,1,2,3 \) to further reduce the number of arithmetic operations. The coefficients are selected in zigzag pruned way and the computational complexity is compared with that of equal number of block pruned coefficients as specified in [18]. For the specific case, we have compared with nine zigzag pruned coefficients and nine block pruned coefficients. Starting from upper left coefficients, the normalized nine zigzag pruned coefficients, \( \hat{T}_{ij} \)’s from (19) are given as under:

\[
\hat{T}_{i0} = [(A + C) + (E + G) + (I + K) + (M + O)],
\hat{T}_{i1} = [(3B + D) + (3F + H) + (3J + L) + (3N + P)],
\hat{T}_{i0} = [3E(G) - (A + C) + (M + O) - (I + K)],
\hat{T}_{i0} = [(A + C) + (E + G) - (3J + L) + (M + O)],
\hat{T}_{i1} = [3J + (F + H) - (3B + D) + (3N + P) - (3J + L)],
\hat{T}_{i0} = [(B - 3D) + (F - 3H) + (J - 3L) + (N - 3P)],
\hat{T}_{i0} = [3(-A - C) + (E - G) - (I - K) + (M - O)],
\hat{T}_{i0} = [(3B + D) + (3F + H) - (3J + L) + (3N + P)]
\] (20)

where,

\[
A = f(0,3) + f(0,0), B = f(0,3) - f(0,0),
C = f(0,2) + f(0,1), D = f(0,2) - f(0,1),
E = f(3,3) + f(3,0), F = f(3,3) - f(3,0),
G = f(3,2) + f(3,1), H = f(3,2) - f(3,1),
I = f(1,3) + f(1,0), J = f(1,3) - f(1,0),
K = f(1,2) + f(1,1), L = f(1,2) - f(1,1),
M = f(2,3) + f(2,0), N = f(2,3) - f(2,0),
O = f(2,2) + f(2,1), P = f(2,2) - f(2,1).
\] (21)

The nine block pruned coefficients are given as:

\[
T_{i0} = x^2[(A + C) + (E + G) + (I + K) + (M + O)],
T_{i1} = x^2y[(3B + D) + (3F + H) + (3J + L) + (3N + P)],
T_{i0} = x^2[(A + C) + (E - G) + (I - K) + (M - O)],
T_{i0} = x^2y[3E(G) - (A + C) + (M + O) - (I + K)],
T_{i1} = x^2y[3J + (F + H) - (3B + D) + (3N + P) - (3J + L)],
T_{i2} = x^2y[3(-A - C) + (E - G) - (I - K) + (M - O)],
T_{i0} = x^2[(A + C) + (E + G) - (I + K) + (M + O)],
T_{i1} = x^2y[(3B + D) + (3F + H) - (3J + L) + (3N + P)],
T_{i2} = x^2y[3(-A - C) + (E - G) - (I - K) + (M - O)].
\] (22)

The expressions for all the coefficients are same as that of zigzag pruned coefficients defined above except that they will be multiplied with the corresponding terms in scaling matrix from (19).

V. COMPUTATIONAL COMPLEXITY COMPARISON

The proposed zigzag pruned DTM algorithm is compared with the algorithms proposed in [15]-[17], recently proposed block pruned algorithm [18] and the traditional separability-symmetry property. A comparison between above algorithms is shown in Table 1. It has been observed that our zigzag pruned algorithm gives lower computation complexity than other methods. Using the proposed method, a considerable saving in hardware resources can be obtained because multipliers are replaced with shift and addition-operations.
For a $n \times n$ pruned block, we need $n^2$ coefficients for image reconstruction. Therefore, it is obvious that comparison should be made between $n \times n$ block pruned with $n^2$ zigzag pruned.

From Table 1, it is observed that our zigzag pruned algorithm gives lower computation complexity than other algorithms. Specifically comparing with recently proposed block pruned method [18], our algorithm gives lower computational complexities for any pruned sizes. By using only one coefficient (dc component) we need 15 additions to compute $\hat{T}_{00}$. For 4-coefficients zigzag prune size, we need 39 additions and 5 shift operations, in contrast with 2×2 prune size which needs 2- multiplications, 39 additions and 7 shift operations. Similarly for 9-coefficient zigzag prune size our algorithm needs 66 additions and 11 shift operations compared to 6-multiplications, 66 additions and 14 shift operations in 3×3 block pruned algorithm. A substantial reduction in computational complexities is achieved because we have normalized the coefficients by merging the multiplication terms with the quantization matrix.

It is also clear that proposed algorithm complexity is much better than that of algorithm in [17] for 2×2 pruned block. Furthermore, the DTM algorithm presented in [16] is a full 4×4 DTM algorithm which is competitive with our 16-coefficient zigzag pruned algorithm. By merging scaling matrix in (19) with the quantization matrix not only reduce the computation but also speed-up the processing.

VI. SIMULATION RESULTS AND ANALYSIS

The proposed algorithm is tested for compression on a set of standard images as shown in Fig. 4. In Fig. 5, a comparison of reconstructed image quality (PSNR in dB) is made between block-pruned sizes of 1-point (dc component), 2×2 3×3 and 4×4 with that of 1-point (dc component), 4, 9 and 16-coefficients zigzag pruned sizes respectively. It has been observed that, 9-coefficients zigzag pruned DTM always gives a higher PSNR than that of its 3×3 block-pruned counterpart, for images, (a) Lena (b) Barbara and (c) Crowd as shown Fig. 5. For instance, Lena image shows almost 0.5 dB improvement, Barbara image shows a 1.8 dB improvement and Crowd image shows a 1.5 dB PSNR improvement in PSNR. But in case of Mandril image, zigzag pruned method shows a marginal of nearly 0.2 dB reduction. We have tested around 25 various kinds of natural and synthetic images. The nine coefficient zigzag pruned DTM shows a higher PSNR than 3×3 block pruned DTM for almost all the images.

VI. CONCLUSIONS

In this paper, a novel fast algorithm of 2-D 4×4 DTM has been proposed which pruned the coefficients in a zigzag fashion. The nine coefficient zigzag pruned DTM always shows better PSNR than 3×3 block pruned DTM. This zigzag order pruning can be more suitable for still images and video coding applications because of considerable improvement in image quality. Furthermore, fast algorithm of different zigzag prune lengths also has been proposed. By pruning the coefficients according to the energy distribution of the transformed image we can speed up the processing while retaining the image quality.

Fig. 4. Original images used for experiment (a) Lena, (b) Barbara, (c) Crowd, and (d) Mandrill.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0/15/1</td>
<td>0/15/0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24/48/0</td>
<td>2/39/7</td>
<td>0/39/5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>6/66/14</td>
<td>0/66/11</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>64/96/0</td>
<td>32/60/0</td>
<td>0/80/16</td>
<td>-</td>
<td>12/80/20</td>
<td>0/80/16</td>
<td></td>
</tr>
</tbody>
</table>
Fig. 5. PSNR comparison plot between zigzag pruned (zzp pruned) and block pruned (bkp pruned) reconstructed images.

REFERENCES