A Fast Zigzag-Pruned 4×4 DTT Algorithm for Image Compression

RANJAN K. SENAPATI[1], UMESH C. PATI[2], KAMALA K. MAHAPATRA[3],
Department of Electronics and Communication Engineering
National Institute of Technology- Rourkela, Orissa.
INDIA
[1]ranjankumarsenapati@gmail.com
[2]ucpati@nitrkl.ac.in
[3]kkm@nitrkl.ac.in

Abstract: - The Discrete Tchebichef Transform (DTT) is a linear orthogonal transform which has higher energy compactness property like other orthogonal transform such as Discrete Cosine Transform (DCT). It is recently found applications in image analysis and compression. This paper proposes a new approach of fast zigzag pruning algorithm of 4x4 DTT coefficients. The principal idea of the proposed algorithm is to make use of the distributed arithmetic and symmetry property of 2-D DTT, which combines the similar terms of the pruned output. Normalization of each coefficient is done by merging the multiplication terms with the quantization matrix 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 3x3 block pruned case, our method provides lesser computational complexity and has higher peak signal to noise ratio (PSNR). The proposed method is implemented on a Xilinx XC2VP30 FPGA device to show its efficient use of hardware resources.

Key-Words: - Discrete Cosine Transform, Discrete- Tchebichef Transform, Image compression, Peak signal to noise ratio, Zigzag Prune.

1 Introduction

Image transform methods using orthogonal kernel functions are commonly used in image compression. One of the most widely known image transform method, Discrete cosine transform (DCT) has been adopted in the standards for still and moving picture coding [1], [2].This happens because it performs much like the statistically optimal Karhunen-Loeve transform under a variety of criteria [1]. The computing devices such as Personal Digital Assistants (PDAs), Digital cameras and mobile phones require a lot of image transmission and processing. Therefore it is essential to have efficient image compression techniques which could be scalable and portable to these smaller computing devices. The Tchebichef moment compression, that is proposed in this paper, is meant for smaller computing devices [3]. The efficiency of Tchebichef moment compression is higher than that of DCT in terms of compression performance for class of images having high intensity variations.

DTT has lower complexity since it requires the evaluation of only algebraic expressions; whereas certain implementation of DCT requires special algorithms or lookup tables for computation of trigonometric functions [4]. In present days the coding standard recommended by ITU-T and MPEG, H.264/MPEG-4 AVC employs a 4×4 integer cosine transform (ICT) due to its low complexity [5]. 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. [6]. Feig and Winograd [7] proposed another fast algorithm for direct cosine transform. These conventional direct methods are row-column methods which computes N point 1-D DCTs for both row and column directions to evaluate 2N sets of data points. Vetterli [8] 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-up is also possible.

Several algorithms for pruning the 1-D DCT in [9]-[13] and 2-D DCT in [14]-[17] has been addressed. In [10], the output-pruned DCT and DST ware computed by slightly modifying output-pruned FFT algorithms for real valued data of the same

size. A recursive pruned DCT algorithm has been presented in [11] with a structure that allows the generation of next higher order pruned DCT from two identical lower order pruned DCTs. In [12] a fast pruning algorithm is proposed which compute N_0 lowest frequency components of length-N discrete cosine transform, where N_0 is any integer. A generalized output pruning algorithm for matrix – vector multiplications is proposed in [13], which eliminate thoroughly the unnecessary operations for computing an output pruning DCT. Peng [14] 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 [15] 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. In [16], an in-place decimation-in-space (DIS) vector-radix fast cosine transform is presented and two pruning algorithms are derived. The first pruning algorithm discusses the computation of $N_0 \times N_0$ out of $N \times N$ DCT points, where both N_0 and N are powers of 2. The second one presents a recursive pruning method for computation of any number of points for arbitrary shaped regions. The two pruning algorithms are compared with row-column approach in terms of computational complexities. The pruning algorithm in [17] makes use of the algorithm in [7] for any number of low-frequency components. The pruning algorithm in [15] and [17] 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 [10] and [14], a sub-block of coefficients out of $N \times N$ is computed.

The Discrete Tchebichef Transform (DTT) is another linear orthonormal version of orthogonal Tchebichef polynomials, that has very similar energy compactness for natural and artificial images. DTT can be also utilized in image feature extraction and pattern recognition [18]. Recently, 4×4 DTT fast algorithms for image compression have been proposed [21]-[24]. A 2×2 block pruned out of 4×4 DTT algorithm which computes the upper left quarter of 4×4 image blocks is proposed in [23].In [24], Saleh proposed a fast 4×4 algorithm suitable for different block sizes.

Having surveyed on different DCT pruning algorithms, we presume that, zigzag pruning algorithm can be a potential candidate for image/video coding applications. In this paper we propose a fast zigzag pruning DTT algorithm of different prune lengths. A comparison with the existing DTT fast algorithms available in the literature till date is made. Finally the reconstructed image quality of different pruned length is evaluated both subjectively and objectively.

The remainder of the paper is organized as follows: Section 2 focuses the mathematical formulation of Discrete Tchebichef algorithm. Comparison of the similar properties between DTT and DCT is made in Section 3. Section 4 presents the proposed zigzag pruning algorithm. The computational complexities of the proposed algorithm are compared with other algorithms in Section 5. Section 6 shows the hardware implementation of DTT algorithms in a Xilinx XC2VP30 device. Section 7 demonstrates the PSNR reconstruction of different standard images using block pruned DTT and zigzag- pruned DTT. Finally Section 8 provides the conclusion and future work.

2 The Discrete Tchebichef Transform

The Discrete Tchebichef Transform (DTT) is relatively a new transform that uses the Tchebichef moments to provide a basis matrix. As with DCT, the DTT is derived from the orthonormal Tchebichef polynomials. This leads to presume that it will exhibit similar energy compaction properties [19].

For a 2-D image function $f(x,y)$ on the discrete domain of $[0,N-1]\times[0,M-1]$, the discrete forward Tchebichef Transform of order $(p+q)$ is defined as

$$T_{pq} = \frac{1}{\tilde{\rho}(p,N)\tilde{\rho}(q,M)} \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} \tilde{t}_p(x)\tilde{t}_q(y) f(x,y)$$
$$p,\ x = 0.....N-1, \qquad (1)$$
$$q,\ y = 0.....M-1.$$

Given a set of Tchebichef transform T_{pq} for a digital image $f(x,y)$, the inverse transformation of Tchebichef moment can be defined as

$$f(x,y) = \sum_{p=0}^{N-1}\sum_{q=0}^{M-1} T_{pq}\tilde{t}_p(x)\tilde{t}_q(y)$$
$$p = 0,1,.....N-1, \qquad (2)$$
$$q = 0,1,.....M-1.$$

The scaled Tchebichef polynomials $\tilde{t}_p(x)$ are defined using the following recurrence relation [20]:

$$\tilde{t}_p(x) = \frac{(2p-1)\tilde{t}_1(x)\tilde{t}_{p-1}(x) - (p-1)\left(1-\frac{(p-1)^2}{N^2}\right)\tilde{t}_{p-2}(x)}{p}$$
$$p = 0,1,.....N-1 \quad (3)$$

where

$$\tilde{t}_0(x) = 1,$$
$$\tilde{t}_1(x) = \frac{2x+1-N}{N}$$

The definition as specified above uses the following scale factor [3] for the polynomial of degree p as

$$\beta(p,N) = N^p. \quad (4)$$

The set $\{\tilde{t}_i(x)\}$ has a squared-norm given by

$$\tilde{\rho}(p,N) = \sum_{i=0}^{N-1}\{\tilde{t}_i(x)\}^2$$
$$= \frac{N\left(1-\frac{1}{N^2}\right)\left(1-\frac{2^2}{N^2}\right).....\left(1-\frac{n^2}{N^2}\right)}{2n+1}. \quad (5)$$

The values of the squared-norm affect the magnitudes of the corresponding moments T_{pq} (1). As specified in [3], the computation of T_{pq} can lead to erroneous results when N is large. This problem can be solved by constructing orthonormal versions of Tchebichef polynomials by modifying the scale factor in (4) as

$$\beta(p,N) = \sqrt{\frac{N(N^2-1)(N^2-2^2).....(N^2-n^2)}{2n+1}}. \quad (6)$$

By denoting the new set of polynomials with the above scale factor as $\{t_i\}$, the recurrence relation given in (3) can change to the following

$$t_p(x) = (A_1 x + A_2)t_{p-1}(x) + A_3 t_{p-2}(x). \quad (7)$$

where $p = 2, 3.....N-1;\ x = 0,1,.....N-1$ and A_1, A_2 and A_3 are as follows:

$$A_1 = \frac{2}{p}\sqrt{\frac{4p^2-1}{N^2-p^2}},$$
$$A_2 = \frac{1-N}{p}\sqrt{\frac{4p^2-1}{N^2-p^2}},$$
$$A_3 = \frac{p-1}{p}\sqrt{\frac{2p+1}{2p-3}}\sqrt{\frac{N^2-(p-1)^2}{N^2-p^2}}. \quad (8)$$

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

$$t_0(x) = \frac{1}{\sqrt{N}},$$
$$t_1(x) = (2x+1-N)\sqrt{\frac{3}{N(N^2-1)}}. \quad (9)$$

Denoting the squared norm by $\tilde{\rho}(p,N)$, so that

$$\tilde{\rho}(p,N) = \sum_{i=0}^{N-1}\{t_p(i)\}^2 = 1.0 \quad (10)$$

The moment equation in (1) now reduce to

$$T_{pq} = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} t_p(x)t_q(y)f(x,y)$$
$$p = 0,1,2.....N-1; q = 0,1,2.....M-1. \quad (11)$$

The inverse moment transform becomes

$$f(x,y) = \sum_{p=0}^{N-1}\sum_{q=0}^{M-1} T_{pq}t_p(x)t_q(y).$$
$$x = 0,1,2.....N-1; y = 0,1,2.....M-1. \quad (12)$$

Equation (12) can also be expressed using a series representation involving matrices as follows

$$f(i,j) = \sum_{p=0}^{N-1}\sum_{q=0}^{M-1} T_{pq}G_{pq}(i,j)$$
$$i = 0,1,.....N-1; \quad j = 0,1.....M-1 \quad (13)$$

where, G_{pq} is called basis image. Assuming equal image dimension $N = M = 8$, the basis image G_{pq} is defined as:

$$G_{pq}=\begin{bmatrix} t_p(0)t_q(0) & t_p(0)t_q(1) & \cdots & t_p(0)t_q(7) \\ t_p(1)t_q(0) & t_p(1)t_q(1) & \cdots & t_p(1)t_q(7) \\ \vdots & \vdots & \cdots & \vdots \\ t_p(7)t_q(0) & \vdots & \cdots & t_p(7)t_q(7) \end{bmatrix} \quad (14)$$

3 Similarity in Properties between DTT and DCT

3.1 Separability

The definition of DTT can be written in separable form as:

$$T_{pq}=\sum_{x=0}^{N-1} t_p(x)\sum_{y=0}^{M-1} t_q(y)\,f(x,y) \quad (15)$$

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

$$g_q(x)=\sum_{y=0}^{M-1} t_q(y)\,f(x,y) \quad (16)$$

$$T_{pq}=\sum_{x=0}^{N-1} t_p(x)\,g_q(x). \quad (17)$$

The transform equation of DCT can be expressed as:

$$C(u,v)=\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\cos\left[\frac{\pi(2x+1)u}{2N}\right]\sum_{y=0}^{M-1}\cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

where

$$\alpha(u)\alpha(v)=\begin{cases}\sqrt{\frac{1}{N}} & for\ u,v=0\\ \sqrt{\frac{2}{N}} & otherwise\end{cases} \quad (18)$$

From Equation (17) and (18) it is clear that 2-D DTT and 2-D DCT are just one dimensional DTT and DCT applied twice by successive 1-D operations, once in x-direction, and once in y-direction.

3.2 Even Symmetry

From [19], it can be shown that Tchebichef polynomials satisfy the property

$$t_p(N-1-x)=(-1)^p t_p(x),\quad p=0,1,....N-1. \quad (19)$$

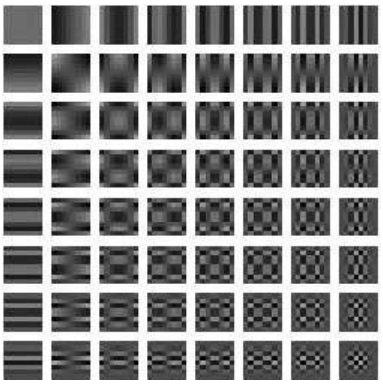
For DCT:

$$C_m(n)=(-1)^m C_m(N-n-1),\quad m=0,1,....N-1. \quad (20)$$

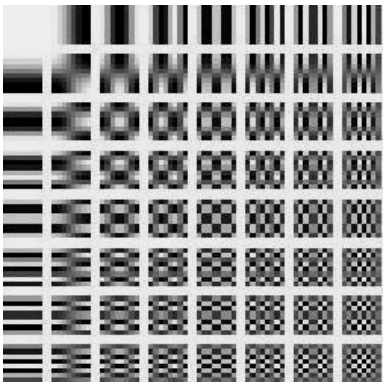
The above two properties are commonly used in transform coding methods to get substantial reduction in the number of arithmetic operations.

3.3 Orthogonality

DTT and DCT basis functions are orthogonal. Thus, the inverse transformation matrix of A is equal to its transpose. Therefore, this property renders some reduction in the pre-computation complexity.2-D basis images of DTT and DCT are shown in Fig 1.



(a)



(b)

Fig. 1. Basis images of (a) DTT (b) DCT

From Fig 1, it is clear that the basis images of DTT and DCT are quite similar in nature. Rows in the

spectrum are increase in horizontal frequencies while columns are increase in vertical frequencies. For both images low frequencies resides in the upper part of spectrum.

3.4 Energy Compaction

Efficiency of a transformation scheme can be gauged by its ability to pack input energy into as many few coefficients as possible. Further, the quantizer discard coefficients with relatively small amplitudes without introducing visual distortion in the reconstructed image. DTT and DCT exhibit excellent energy compaction properties for highly correlated images. The energy of the image is packed into low frequency region i.e. top left region.

4 Proposed Zigzag Pruned 4×4 DTT-Algorithm

The 2-D DTT can be expressed in matrix form as

$$T = \tau F \tau'. \tag{21}$$

where F is the 2-D input data, τ is the Tchebichef basis and T is the 2-D matrix of transformed coefficients. The transform kernel for 4 point DTT can be defined from (11) as

$$\tau = \begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ -3/2\sqrt{5} & -1/2\sqrt{5} & 1/2\sqrt{5} & 3/2\sqrt{5} \\ 1/2 & -1/2 & -1/2 & 1/2 \\ -1/2\sqrt{5} & 3/2\sqrt{5} & -3/2\sqrt{5} & 1/2\sqrt{5} \end{bmatrix} \tag{22}$$

By defining $x = 1/2$ and $y = 1/\sqrt{5}$, (22) can be written as:

$$\tau = \begin{bmatrix} x & x & x & x \\ -3xy & -xy & xy & 3xy \\ x & -x & -x & x \\ -xy & 3xy & -3xy & xy \end{bmatrix} \tag{23}$$

Factorizing (23) we will get

$$S = \begin{bmatrix} x & x & x & x \\ xy & xy & xy & xy \\ x & x & x & x \\ xy & xy & xy & xy \end{bmatrix}, \hat{\tau} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -3 & -1 & 1 & 3 \\ 1 & -1 & -1 & 1 \\ -1 & 3 & 3 & 1 \end{bmatrix}. \tag{24}$$

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

The expression in (21) can be factorized as:

$$T = (\hat{\tau} F \hat{\tau}') \otimes \hat{S}, \tag{25}$$

where,

$$\hat{S} = \begin{bmatrix} x^2 & x^2 & x^2 & x^2 \\ x^2y & x^2y^2 & x^2y & x^2y^2 \\ x^2 & x^2y & x^2 & x^2y \\ x^2y & x^2y^2 & x^2y & x^2y^2 \end{bmatrix} \quad \text{and} \quad \hat{T} = \hat{\tau}\, F\, \hat{\tau}.$$

Symbol ⊗ 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 (23) in (21), 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 a zigzag pruned way and the computational complexity is compared with that of equal number of block pruned coefficients as specified in [24]. 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 (25) are given as:

$$\begin{aligned}
\hat{T}_{00} &= [(A+C)+(E+G)+(I+K)+(M+O)]. \\
\hat{T}_{01} &= [\{(3B+D)+(3F+H)\}+\{(3J+L)+(3N+P)\}]. \\
\hat{T}_{10} &= [3\{(E+G)-(A+C)\}+(M+O)-(I+K)]. \\
\hat{T}_{20} &= [(A+C)+(E+G)-\{(I+K)+(M+O)\}]. \\
\hat{T}_{11} &= [3(3F+H)-3(3B+D)+(3N+P)-(3J+L)]. \\
\hat{T}_{02} &= [(A-C)+(E-G)+(I-K)+(M-O)]. \\
\hat{T}_{03} &= [(B-3D)+(F-3H)+(J-3L)+(N-3P)]. \\
\hat{T}_{12} &= [3\{-(A-C)+(E-G)\}-(I-K)+(M-O)]. \\
\hat{T}_{21} &= [\{(3B+D)+(3F+H)\}-\{(3J+L)+(3N+P)\}].
\end{aligned} \tag{26}$$

where,

$$\begin{aligned}
&A = f(0,3)+f(0,0),\ B = f(0,3)-f(0,0), \\
&C = f(0,2)+f(0,1),\ D = f(0,2)-f(0,1), \\
&E = f(3,3)+f(3,0),\ F = f(3,3)-f(3,0), \\
&G = f(3,2)+f(3,1),\ H = f(3,2)-f(3,1), \\
&I = f(1,3)+f(1,0),\ J = f(1,3)-f(1,0), \\
&K = f(1,2)+f(1,1),\ L = f(1,2)-f(1,1), \\
&M = f(2,3)+f(2,0),\ N = f(2,3)-f(2,0), \\
&O = f(2,2)+f(2,1),\ P = f(2,2)-f(2,1),
\end{aligned} \tag{27}$$

Fig 2 shows the signal flow graph of nine zigzag pruned coefficients.

The nine normalized block pruned coefficients are given as:

$$\hat{T}_{00},\ \hat{T}_{01},\ \hat{T}_{02},\ \hat{T}_{10}, \hat{T}_{11}, \hat{T}_{12}, \hat{T}_{20}, \hat{T}_{21}, \hat{T}_{22}. \quad (28)$$

The expressions for all the coefficients are same as that of zigzag pruned coefficients defined in (26), except, coefficient $\hat{T}_{22}$ which is can be expressed as:

$$\hat{T}_{22} = [\{(A-C)+(E-G)\}-\{(I-K)+(M-O)\}]. \quad (29)$$

5 Computational Complexity Analysis

The proposed zigzag pruned DTT algorithm is compared with the recently proposed algorithms in [21]-[24] and the traditional separability-symmetry algorithm. 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.

Table 1 shows that our zigzag pruned algorithm gives lower computation complexity than other algorithms. Specifically comparing with recently proposed block pruned method [24], our algorithm has 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 pruned size, we need 39 additions and 5 shift operations, as compared with 2×2 pruned size which needs 2- multiplications, 39 additions and 7 shift operations. Similarly for 9 coefficient pruned 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. This is due to the fact that, we have normalized the coefficients by merging the multiplication terms with the quantization matrix.

It is also clear that proposed algorithm complexity is lower than that of algorithm in algorithm in [23] for 2×2 pruned block. The DTT algorithm presented in [22] is a full 4×4 DTT algorithm which is having same complexity as the full 16-coefficient zigzag algorithm.

6 Hardware Implementation

The proposed algorithm is implemented on Xilinx XC2VP30 FPGA device. We developed a distributed arithmetic based approach to compute 1-D and 2-D DTT transform on Xilinx XC2VP30 platform. This is due to the fact that, DA [25] is free from multiplications. All the coefficients are determined by integer shift and addition operations.

Table 1 shows the hardware resource utilization of 1-D floating point DTT algorithm. The number of slices and 4 input LUTs are 2% and 1.2% of the available resources. This is much higher than 1D integer DTT as shown in table 4.

Considering the case of 2D DTT, it can be seen from table 3 that, pruned DTT do not require any flip flops (memory) in contrast to 2D DTT in Table 2, which requires almost 1% of the resources. This makes the transform a combinational circuit, instead of a sequential one. This is a major advantage of using pruning method. Further, the number of slices and 4 input LUTs are 2.3% and 2% lesser in 9-pruned DTT than 2D float point DTT. Number of bonded IOBs are 37% more in pruned DTT, which is the only drawback.

Table 2 and table 5 show the resource utilization summery of floating point and integer based 2D DTT. By looking at table 5, it can be said that, using integer based transform much less hardware resources can be saved. The pruned DTT is calculated using direct approach rather than row-column approach. The merit of direct approach in calculating transform is that, it does not need any memory elements, which is obvious in table 3. Figure 2 shows the signal flow graph of 9- pruned normalized coefficients. The values of A,B,..P are the values obtained from equation (7). In the signal flow graph, multiplications with 3 are implemented as a left shift and add operations.

Table 1. H/W utilization of 1D floating point DTT in Xilinx XC2VP30.

Resources	Available	Utilise	% Utilisation
No of Slices	13696	277	2
Flip Flops	27392	0	0
4 input LUTs	27392	493	1.2
Bonded IOBs	556	76	13

Table 2. H/W utilization of 2D floating point DTT on Xilinx XC2VP30

Resources	Available	Utilise	% Utilisation
No of Slices	13696	707	5
Flip Flops	27392	204	0.9
4 input LUTs	27392	1286	4
Bonded IOBs	556	59	10

Table 3. H/W utilization of pruned DTT on Xilinx XC2VP30 device.

Resources	Available	Utilise	% Utilisation
No of Slices	13696	380	2.7
Flip Flops	27392	0	0
4 input LUTs	27392	715	2
Bonded IOBs	556	263	47

Table 4 H/W utilization of 1D Integer DTT on Xilinx XC2VP30 device.

Resources	Available	Utilise	% Utilisation
No of Slices	13696	61	0.4
Flip Flops	27392	0	0
4 input LUTs	27392	112	0.4
Bonded IOBs	556	84	11

Table 5. H/W utilization 2D integer DTT on Xilinx XC2VP30 device.

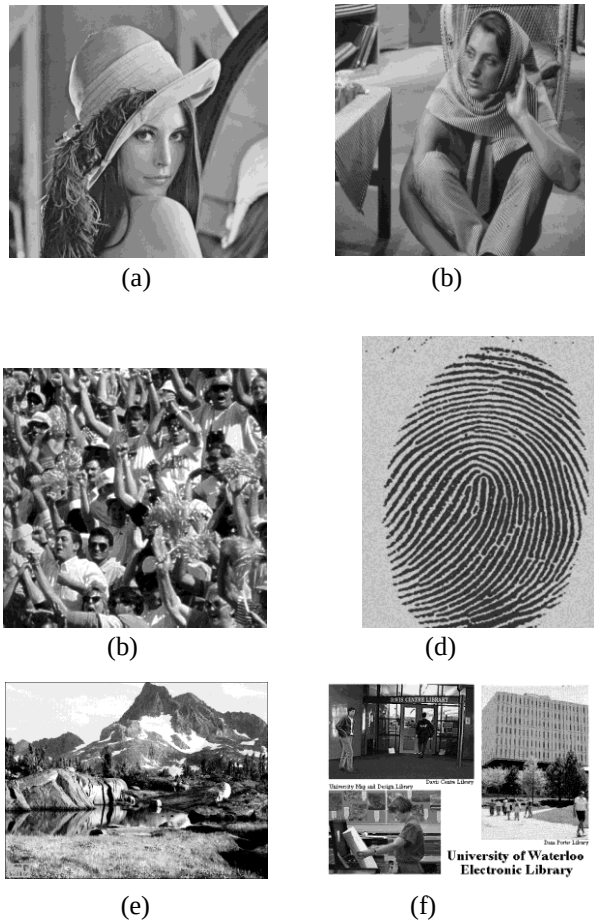
Resources	Available	Utilise	% Utilisation
No of Slices	13696	98	0.7
Flip Flops	27392	94	0.3
4 input LUTs	27392	157	0.6
Bonded IOBs	556	63	11

7 Results and Discussion of Block-Pruned and Zigzag-Pruned DTT/DCT

The proposed algorithm is tested for compression on a set of standard images. A comparison of reconstructed image quality (PSNR in dB) is made between block-pruned sizes of dc component, 2x2 and 3x3 with that of dc component, 4 points and 9 points zigzag pruned sizes respectively. Table 7 Shows the PSNR comparison between block pruned DTT and zigzag pruned DTT. From the Table 7(a) and (b), it can be observed that the PSNR of reconstructed images using 9- coefficients zigzag pruned sizes are higher than that of the PSNR of 3×3 block-pruned image sizes. Comparing with 4-coefficients zigzag pruned sizes and 2×2 block pruned sizes, the PSNR of block pruned sizes are higher than that of zigzag pruned sizes. Nevertheless, there is advantage of computational complexities in both cases. Similarly, in table 8(a) and (b) an exhaustive comparison is made between block-pruned DTT with block-pruned DCT and zigzag-pruned DTT with zigzag-pruned DCT. It has seen observed that, 9- coefficients zigzag pruned DTT/DCT shows always a higher PSNR than that of its 3×3 block-pruned counterpart. For example, in case of Lena image DTT shows a PSNR gain of 0.7 dB, Barbara shows a significant gain of 1.54 dB, for crowd image DTT shows a PSNR gain of 1.02 dB. Similar performance improvement is also present in DCT 9-pruned images. Furthermore, for images such as (a) Lena, (b) Barbara and (c) Crowd, DCT shows slight better performance than that of DTT. For images (d) Finger print, (e) Mountain and (f) Library, DTT outperforms DCT of any pruning sizes. For instance, Finger print image shows a PSNR gain of 1.27 dB, in 9-prune sizes and 0.27 dB in 4- prune sizes. For Mountain and Library images the PSNR gain is only 0.02 to 0.03 dB. Hence, 9-coefficients pruned sizes are enough for practical image or video coding applications.

8 Conclusion

In this paper, a fast algorithm of 2-D 4×4 DTT has been proposed which pruned the coefficients in a zigzag fashion. This zigzag order pruning can be more suitable for still images and video coding applications because of considerable improvement in objective image quality and fast processing. The pruning algorithm is implemented in a Xilinx XC2VP30 FPGA, which shows considerable amount of hardware savings than a 4x4 floating point DTT. Furthermore, it has been shown that DTT compression is very similar to DCT compression for natural and artificial images. Future research direction is to develop a fast 8x8 DTT algorithm for real time image and video compression.



(a) (b) (b) (d) (e) (f)

Fig. 2. Set of original images used for experimental evaluation of the Pruned DTT/DCT, Zigzag DTT/DCT. (a) Lena, (b) Barbara, (c) Crowd, (d) Finger print, (e) Mountain, (f) Library

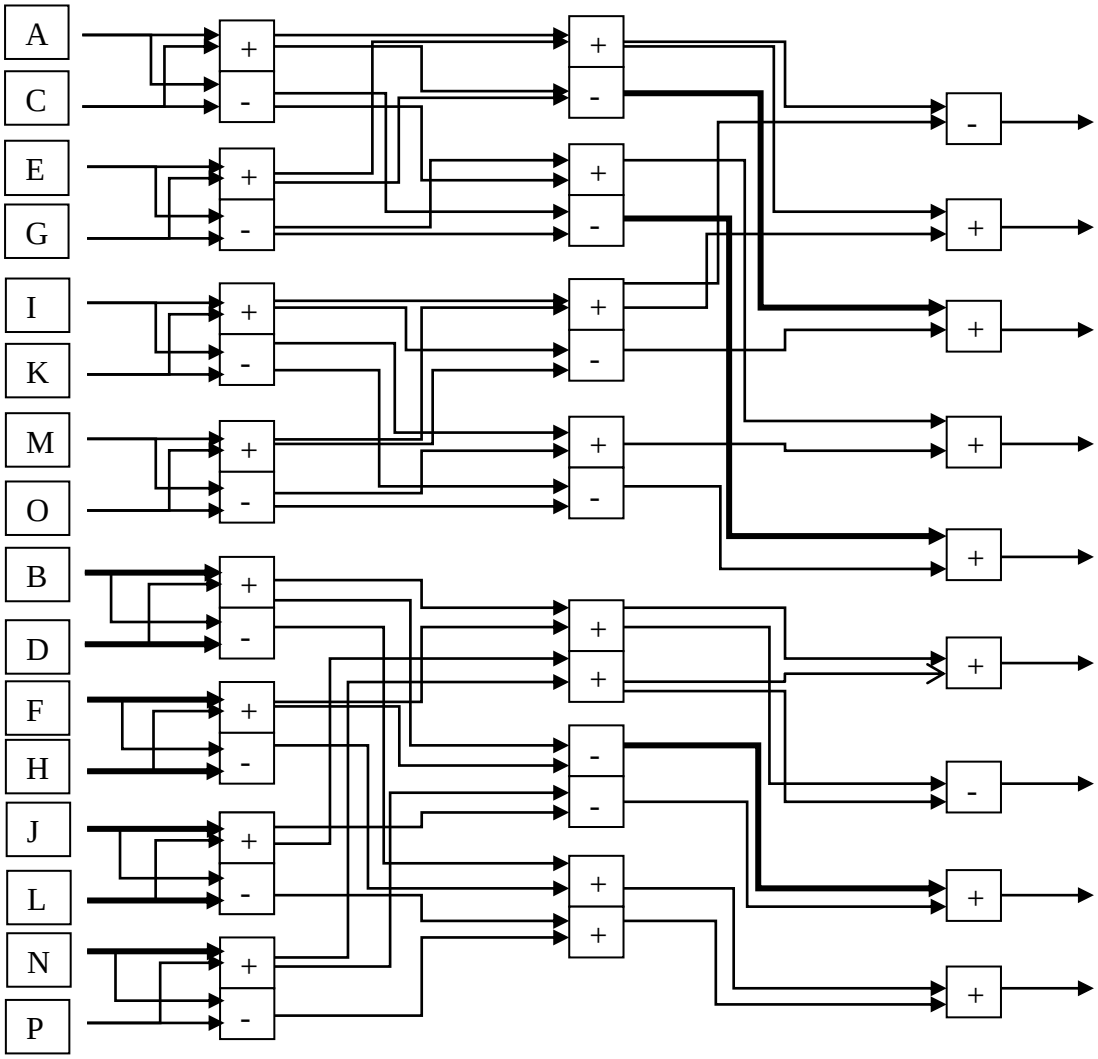


Fig 3. Signal flow graph of 9 zigzag prune normalized coefficients

Table 6. Computational complexity comparison between different DTT algorithms and our proposed algorithm

No. of coefficients used for DTT image reconstruction (Pruned DTT)	Number of operations Multiplications/additions/shift					
	Separability & Symmetry	Nakagaki & Mukundnan[21]	Ishwar et. al.[22]	Abdelwaheb [23]	Block Pruned Method[24]	Proposed method
1	-	-	-	-	0/15/1	0/15/0
4	-	-	-	24/48/0	2/39/7	0/39/5
9	-	-	-	-	6/66/14	0/67/11
16	64/96/0	32/60/0	0/80/16	-	12/80/20	0/80/16

Table 7(a): Comparison of PSNR between block-pruned and zigzag-pruned reconstructed images of (a) Lena, (b) Barbara and (c) Crowd

Number of DTT Coefficients retained	PSNR(dB)					
	Lena		Barbara		Crowd	
	Block prune	Zigzag prune	Block pruned	Zigzag prune	Block prune	Zigzag prune
1	26.92	26.92	23.37	23.37	21.62	21.62
4	33.35	32.35	25.61	25.23	30.13	29.43
9	39.29	40.00	30.03	31.57	38.54	39.59

Table 7(b): Comparison of PSNR between block-pruned and zigzag-pruned reconstructed images of (d) Finger print, (e) Mountain and (f) Library

Number of DTT Coefficients retained	PSNR(dB)					
	Finger print		Mountain		Library	
	Block prune	Zigzag prune	Block prune	Zigzag prune	Block prune	Zigzag prune
1	11.08	11.08	17.08	17.08	16.25	16.25
4	14.94	16.84	19.60	19.82	18.90	19.44
9	22.28	24.38	22.97	23.17	22.69	23.45

Table 8(a): Comparison of PSNR between DCT and DTT of block pruned and zigzag pruned reconstructed images of (a) Lena, (b) Barbara and (c) Crowd

Number of coefficients retained for image reconstruction	PSNR(dB)											
	Lena				Barbara				Crowd			
	Block prune		Zigzag prune		Block prune		Zigzag prune		Block prune		Zigzag prune	
	DCT	DTT	DCT	DTT	DCT	DTT	DCT	DTT	DCT	DTT	DCT	DTT
1	26.92	26.92	26.92	26.92	23.37	23.37	23.37	23.37	21.62	21.62	21.62	21.62
4	33.43	33.36	32.38	32.35	25.68	25.61	25.29	25.23	30.20	30.12	29.47	29.43
9	39.65	39.29	40.24	39.99	30.29	30.03	31.78	31.57	39.15	38.54	40.03	39.59

Table 8(b): Comparison of PSNR between DCT and DTT of block pruned and zigzag pruned reconstructed images of (d) Finger print, (e) Mountain and (f) Library

Number of coefficients retained for image reconstruction	PSNR(dB)											
	Finger Print				Mountain				Library			
	Block prune		Zigzag prune		Block prun		Zigzag prune		Block prune		Zigzag prune	
	DCT	DTT	DCT	DTT	DCT	DTT	DCT	DTT	DCT	DTT	DCT	DTT
1	21.88	21.88	21.88	21.88	17.07	17.07	17.07	17.07	16.25	16.25	16.25	16.25
4	28.05	28.25	29.11	29.38	19.59	19.59	19.80	19.82	18.90	18.90	19.42	19.44
9	30.42	30.75	31.94	33.21	22.96	22.97	23.16	23.17	22.69	22.69	23.42	23.45

References:

[1] K.R.Rao and P.Yip(1990):*Discrete cosine transform algorithms, advantages, applications*, Academic Press Inc.

[2] G.K.Wallace(1991): *The JPEG still picture compression standard, Communications of the ACM*, 34(4), 30-44.

[3] R.Mukundan,"Some Computational aspects of Discrete Orthonormal Moments," *IEEE Transaction on Image Processing.* vol. 13, no.8, pp.1055-1059, Aug 2004.

[4] Robert Kutka,"Fast computation of DCT by statistic adapted look-up tables", In Proc. of *IEEE International Conference on Multimedia and Expo.* (ICME) 2002, vol.1, pp. 781-784.

[5] H.S.Malvar, .Hallapuro, M.Karczewicz and L.Kerofsky, "Low-complexity Transform and Quantization in H.264/avc."*IEEE Trans. Circuits Syst.*, vol.13, pp.598-603, Jul 2003.

[6] P.Duhamel and C.Guillemot, "Polynomial transform computation of 2-D DCT," Proc. *ICASSP'90*, pp 1515-1518, 1990.

[7] E.Feig and S,Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Processing*. vol.40,no.9, pp. 2174-2193, Sep.1992.

[8] M.Vetterli, "Fast 2-D discrete cosine transform", Proc. *IEEE ICASSP'85*, pp.1538-1541, 1985.

[9] Z.Wang, "Pruning the Fast Discrete Cosine Transforms", *IEEE Transactions on Communications*, vol.39, no.5, pp.640-643, May 1991.

[10] R.Stasinski, "On pruning the Discrete Cosine and Sine Transforms" *IEEE MELECON* 2004, Dubrovnik, Croatia, pp. 269-271, May 12-15, 2004.

[11] M.El-Sharkawy and W.Eshmawy, "A fast recursive pruned DCT for image compression applications", *37th Midwest Symposium on Circuits and Systems*, pp. 887-890, 1995.

[12] A.N.Skodras, "Fast discrete cosine transform pruning", *IEEE Trans. on Signal Processing*, vol.48, no.2, pp. 1833-1837, Jul. 1994.

[13] Y.Huang, JWu, and C.Chang, "A generalized output pruning algorithm for matrix-vector multiplication and its application to computing pruning discrete cosine transform", *IEEE Trans. Signal Processing*, vol.48, no2,pp. 561-563, Feb 2000.

[14] S.Peng, "Complexity scalable video coding via IDCT data pruning", Proc *Internal Conference on Consumer Electronics* (ICCE'01), Los Angeles, CA, pp 74-75, Jun 2001.

[15] N.P.Walmsley, A.N.Skodras and K.M.Curtis, "A Fast Picture Compression Technique", *IEEE Trans. on Consumer Electronics*, vol.40, no.1, pp.11-19, Feb 1994.

[16] C.A.Christopoulos, and A.N.Skodras, "Pruning the 2-D fast cosine transform", Proceedings of *VII European Signal Processing Conference* (EUSIPCO), Edinburgh, Scotland, UK, pp.596-599, September 13-16, 1994.

[17] A. Silva and A. Navarro, "Fast 8×8 DCT Pruning Algorithm", *IEEE International Conference on Image Processing* (ICIP), Geneva, Italy, pp. 317-320, Sept 2005.

[18] R.Mukundnan, "Radial Tchebichef invariants for pattern recognition", In Proc. of *IEEE TENCON'05 Conference*, Melboroune, 21-24 Nov. 2005, pp. 2098-2103(1-6).

[19] R.Mukundan, "Image Analysis by Tchebichef Moments", *IEEE Trans on Image Processing*, vol.10,no.9,pp.1357-1364, 2001.

[20] L.Kotoulas and I.Andreadis, "Fast Computation of Chebyshev Moments", *IEEE Trans. on Circuits System for Video Technology*, vol.16,no.7, July 2006.

[21] K.Nakagaki and R.Mukundnan, "A Fast 4×4 Forward Discrete Tchebichef Transform Algorithm", *IEEE Signal Processing Letters*, vol.14, no.10, Oct.2007, pp.684-687.

[22] S.Ishwar, P.K.Meher and M.N.S.Swamy, "Discrete Tchebichef Transform-A Fast 4×4 algorithm and its application in Image/Video Compression", *ISCAS 2008*, May 18-21, pp.260-263, USA 2008.

[23] S.A.Abdelwahab, "Image Compression using Fast and Efficient Discrete Tchebichef Algorithm," In Proc of INFOS 2008, *The 6th International Conference on Informatics and Systems*, Cairo, Egypt, March 27-29, 2008, pp.49-55.

[24] Hassan I.Saleh, "A Fast Block-Pruned 4×4 DTT Algorithm for Image Compression," In *International Journal of Computer Theory and Engineering*, vol.1, no.3, August, 2009, pp. 1793-8201.

[25] Peng Chungan, Y.U. Dunshan and Z. Xing, "A 250 MHz Optimised Distributed Architecture of 2D 8X8 DCT", 7th Internal Conference on ASICs",pp. 189-192, Oct, 2007.