International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:3, No:8, 2009

A Novel Recursive Multiplierless Algorithm for 2-D DCT

V.K.Ananthashayana[I], Geetha.K.S[II]

Abstract**—**In this paper, a recursive algorithm for the computation of 2-D DCT using Ramanujan Numbers is proposed. With this algorithm, the floating-point multiplication is completely eliminated and hence the multiplierless algorithm can be implemented using shifts and additions only. The orthogonality of the recursive kernel is well maintained through matrix factorization to reduce the computational complexity. The inherent parallel structure yields simpler programming and hardware implementation and provides $\frac{3}{2}N\log_2 N-N+1$ additions and $\frac{N}{2}\log_2 N$ shifts which is very much less complex when compared to other recent multiplierless algorithms.

Keywords**—**DCT, Multilplerless, Ramanujan Number, Recursive.

I. Introduction

THE Discrete Cosine Transform (DCT) [1] is widely utilized in many image/video coding applications for the removal of spatial redundancy. Many fast and efficient algorithms for computing 2-D DCT/IDCT have been intensively discussed [2-4] to reduce the computational complexity and to facilitate the real-time implementation. Recently, there has been increasing interest in approximating a given floating-point transform using only Very Large Scale Integration (VLSI) friendly binary, multiplierless coefficients. Since only binary coefficients are needed, the resulting transform approximation is multiplierless, and the overall complexity of hardware implementation can be measured in terms of the total number of adders and/or shifters required in the implementation. The fast bi-orthogonal Binary DCT (BinDCT) [5] and Integer DCT (IntDCT) [6, 7] belong to a class of multiplierless transforms which computes the coefficients of the form $k/2^b$. They compute the integer to integer mapping of coefficients through the lifting matrices. The performances of these transforms depend upon the lifting schemes used and the round off functions. In general, these algorithms require the approximation of the decomposed DCT transformation matrices by proper diagonalisation. Thus the complexity is shifted to the techniques used for decomposition. Among the direct implementations, Wang [8] and Chen [9] have given the fast DCT algorithms whose computational complexity is same. Lee [3] proposed a fast algorithm of IDCT, but the inversion and division of the cosine values give rise to the instability problems. The papers [2, 4 ,8 ,9] have given the fast algorithms of computational complexity $\frac{N}{2}\log_2 N$ multiplications and $\frac{3}{2}N\log_2 N-N+1$ additions. Computation of DCT coefficients involves evaluation of cosine angles of multiples of 2π/N. If N is chosen such that it could be represented as $2^{-l}+2^{-m}$, where l and m are integers, then the trigonometric functions can be evaluated recursively by simple shift and addition operations. Such integers are called Ramanujan numbers[III], defined by Bhatnagar [10, 11] after the great mathematician, S.Ramanujan. Use of Ramanujan Number for computing DCT was outlined by the author in [12]. This paper intends to improvise the computational complexity of the algorithm in our previous work [13] by using the recursive structure to decompose the DCT transformation matrix.

In this paper, we propose a 2-D fast multiplierless discrete cosine transform using Ramanujan Numbers, which computes the coefficients extracting the orthogonal property of the recursive kernel of DCT. Matrix factorization allows us to yield $\frac{N}{2}\log_2 N$ shifts and $\frac{3}{2}N\log_2 N-N+1$ additions. But the proposed algorithm eliminates the evaluation of the floating point cosine values by Ramanujan Numbers and thus the scalar multiplications can be implemented using shifters.

This paper is organized as follows: the Ramanujan Number of order-1 and order-2 is discussed in Section II. Evaluation of the cosine values using Ramanujan Number is explained in section III. The Orthogonal property of the DCT kernel is discussed in Section IV. In Section V, we propose the recursive algorithm to compute 2-D DCT using Ramanujan Number. The experimental results are shown in Section VI, and finally concluding remarks are described in Section VI.

II. Ramanujan Number of Order-1 & Order-2

Ramanujan Numbers are related to π and integers which are powers of 2. It was Ramanujan who determined the value of π to large decimal point accuracy. Ramanujan Number of order-

[I] V.K.Ananthashayana is Prof. & Head of Dept. of CSE, M.S.Ramaiah Institute of Technology, Bangalore, India-560054. (phone: 91-080-23600822; fax: 080-23603124; e-mail: anantha_k2@yahoo.co.in).

[II] Geetha.K.S is with R.V.College of Engineering, Banagalore, India-560059 (e-mail: geethakomandur@gmail.com).

[III] As per our interaction with Bruce Berndt & N.Bhatnagar, it is known that Bhatnagar presented Ramanujan Number of order-1 & order-2.

1 was used in [4, 6] to compute the Discrete Fourier Transform.

A. Definition : Ramanujan Number of order-1

Ramanujan Numbers of order-1 $\Re_1(a)$ are defined as follows:

$$\Re(a) = \left[\frac{2\pi}{l_1(a)}\right] \quad where\ l_1(a) = 2^{-a} \tag{1}$$

a is a non-negative integer and $[\cdot]$ is a round off function. The numbers could be computed by simple binary shifts. Consider the binary expansion of π which is 11.00100100001111… If *a* is chosen as 2, then $\iota_1(2) = 2^{-2}$, and $\Re_1(2) = [11001.001000.......] = 11001$. i.e., $\Re_1(2)$ is equal to 25. Likewise $\Re_1(4)$=101. Thus the right shifts of the decimal point (a+1) time yields $\Re_1(a)$. Ramanujan used these numbers to approximate the value of π. Let this approximated value be $\hat{\pi}$ and let the relative error of approximation be ε, then

$$\hat{\pi} = \frac{1}{2}\left[\Re_1(a)\iota(a)\right] \quad \hat{\pi} = (1+\in)\pi \tag{2}$$

These errors could be used to evaluate the degree of accuracy obtained in computation of DCT coefficients.

B. Definition: Ramanujan Number of order-2

The Ramanujan Number of order-2 [10] are defined such that $2\pi/N$ is approximated by sum or difference of two numbers which are negative powers of 2. Thus Ramanujan Numbers of order-2 are,

$$\Re_{2\iota}(l,m) = \left[\frac{2\pi}{\iota_{2i}(l,m)}\right] \quad for\ i = 1,2 \tag{3}$$

$$\iota_{21}(l,m) = 2^{-l} + 2^{-m} \quad for\ m > l \ge 0 \tag{4}$$

$$\iota_{22}(l,m) = 2^{-l} - 2^{-m} \quad for\ (m-1) > l \ge 0 \tag{5}$$

Where *l* and *m* are integers, Hence

$$\Re_{21}(3,5) = 40 \quad \Re_{21}(1,3) = 10 \tag{6}$$

These higher-order numbers give better accuracy at the expense of additional shifts and additions. Ramanujan Number of order-2 and their properties are listed in the table I below.

TABLE I
RAMANUJAN NUMBER OF ORDER-2

(l,m)	$\Re(l,m)$	$\hat{\pi}$	Upper bound of Error
0,2	5	3.125	4.872×10^{-6}
1,2	8	3.0	5.42×10^{-4}
4,5	67	3.140	6.89×10^{-6}

III. EVALUATION OF COSINE VALUES USING RAMANUJAN NUMBER

Let us define $2\pi/N$ equal to x and c_n equal to $\cos(nx)$. We approximate x as $\hat{x} = 2\hat{\pi}/N$. We can then approximate c_n 's by $z_n(\alpha)$, where $\alpha = \hat{x}^2/2 - \hat{x}^4/4!$. If N is a Ramanujan Number of order-2, then $\alpha = 2^{-l} + 2^{-m}$.

Thus,

$$\begin{aligned} &t_0(\alpha) = 1 \\ &t_1(1) = 1-\alpha, \\ &t_{n+1}(\alpha) = 2(1-\alpha)t_n(\alpha) - t_{n-1}(\alpha) \end{aligned} \tag{7}$$

It can be observed the above recursive equation are closely related to Chebyshev polynomials of the first kind and thus c_n 's are computed using just shift and addition operations.

IV. ORTHOGONAL PROPERTY OF DCT KERNEL

DCT matrix enjoys the property of orthogonality, which follows that the inverse of the matrix equals its transpose. This property allows us to provide new possibilities for those algorithms whose recursive kernel is also orthogonal.

Suppose the matrix factorization form of the recursive kernel of a fast DCT algorithm is:

$$[Q_N] = [A][B]\begin{bmatrix} Q_{\frac{N}{2}} & 0 \\ 0 & Q_{\frac{N}{2}} \end{bmatrix}.[C][D] \tag{8}$$

where [A], [B], [C] & [D] are permutation matrices, integer matrix used for decomposing the DCT kernel.

By applying the transforms of inverse and transposition to both sides of equation (1) we get

$$[Q_N]^{-1*T} = [A]^{-1*T}.[B]^{-1*T}.\begin{bmatrix} Q_{\frac{N}{2}}^{-1*T} & 0 \\ 0 & Q_{\frac{N}{2}}^{-1*T} \end{bmatrix}.[C]^{-1*T}.[D]^{-1*T} \tag{9}$$

Since we know that [Q] is orthogonal, the equation could be written as

$$\left[Q_N\right]^{-1*T} = \left[A\right]^{-1*T} . \left[B\right]^{-1*T} . \begin{bmatrix} Q_{\frac{N}{2}} & 0 \\ 0 & Q_{\frac{N}{2}} \end{bmatrix} . \left[C\right]^{-1*T} . \left[D\right]^{-1*T} \quad (10)$$

The equation explores a new method of factorizing the DCT algorithm. Thus the property of orthogonality provides new possibilities for other algorithms, and also improves the existing algorithms

V. Recursive structure using Ramanujan Numbers

According to the definition of DCT [1] for a given data sequence {x (n); n=0, 1, ,2…N-1}, the DCT coefficient sequence {X (k); k=0, 1, 2 ….N-1} is given by the following relation:

$$C(k) = \frac{2\varepsilon_k}{N} \sum_{n=0}^{N-1} x(n) \cos\left(\pi k \frac{(2n+1)}{2N} \right) \quad (11)$$

for k=0,1,2,……N-1

where

$$\varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & for \;\; k = 0 \\ 1 & otherwise \end{cases}$$

We could represent the DCT kernel as

$$\hat{A}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \quad (12)$$

Where $a = 1/\sqrt{2}$, $b = \cos(\pi/8)$ and $c = \cos(3\pi/8)$.
In general, we could define the matrix of the recursive kernel as C_N, so that we have the DCT Coefficient matrix defined as

$$A_N = \left[P_L\right] . \left[C_N\right] \quad (13)$$

and

$$\left[C_N\right] = \left[P_1\right] . \begin{bmatrix} C_{\frac{N}{2}} & 0 \\ 0 & C_{\frac{N}{2}} \end{bmatrix} . \begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix} . \left[P_2\right] \quad (14)$$

Where

(1) $[P_1]$ is the permutation matrix to interlace the two halves of the input data sequence as $\tilde{x} = \left[P_1\right].x$ where

$$\tilde{x} = \left[x_1, x_{\frac{N}{2}+1}, x_2, x_{\frac{N}{2}+2}, \ldots x_{\frac{N}{2}}, x_N \right]$$

and $x = \left[x_1, , x_2, \ldots . x_{N-1}, x_N \right]$.

(2) $[\boldsymbol{T}]$ is a diagonal trigonometric matrix. For a DCT matrix of length N, $[\boldsymbol{T}]$ is a $M \times M$ matrix, where $M=N/2$.

$$\left[T\right] = diag\left[\cos\left(\theta_m\right)\right], \; \theta_m = \frac{2\pi\left(2m+1\right)}{N} 2^{-2}$$

$$m = 0, 1, \ldots . M - 1$$

N being a Ramanujan Number, the matrix $[\boldsymbol{T}]$ could then be represented as

$$\left[T\right] = diag\left(2^{-l_{m1}} + 2^{-l_{m2}}\right)$$

where l_{m1} and l_{m2} are non-negative integers.

(3) $[P_2]$ is an integer coefficient matrix to perform additions and subtractions of the input sequence. For $\tilde{y} = \left[P_2\right].y$

where $\tilde{y} = \left[y_1 + y_N, y_2 + y_{N-1} \ldots . y_{\frac{N}{2}-1} + y_{\frac{N}{2}+1}, y_1 - y_N, y_2 - y_{N-1} \ldots . y_{\frac{N}{2}-1} - y_{\frac{N}{2}+1} \right]$

(4) $[P_L]_N$ is the product of $\log_2 N - 1$ sparse factor matrices.

$$\left[P_L\right]_N = \left[P_{L(\log_2 N - 1)}\right]_N \ldots . \left[P_{L2}\right]_N . \left[P_{L1}\right]_N$$

For length 2N sequence, we have

$$\left[P_L\right]_{2N} = \left[P_{L(\log_2 N-1)}\right]_{2N} . \begin{bmatrix} \left(P_{L(\log_2 N-1)}\right)_N & 0 \\ 0 & \left(P_{L(\log_2 N-1)}\right)_N \end{bmatrix}_{2N} \ldots \begin{bmatrix} \left(P_{L1}\right)_4 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \left(P_{L1}\right)_4 \end{bmatrix} \quad (15)$$

where

$$\left[P_{L(\log_2 N-1)}\right]_N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 2 & 0 & 0 \\ 0 & 0 & \cdot^{\cdot^{\cdot}} & 0 & 0 & 0 & \ddots & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}_N$$

for ex.

$$\left[P_L\right]_4 = \left[P_{L1}\right]_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 2 \end{bmatrix}$$

$$[P_L]_8 = [P_{L2}]_8[P_{L1}]_8 = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&-1&0&2&0&0\\ 0&0&-1&0&0&0&2&0\\ 0&-1&0&0&0&0&0&2 \end{bmatrix} \cdot \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&-1&0&2&0&0\\ 0&0&-1&0&0&0&2&0\\ 0&-1&0&0&0&0&0&2 \end{bmatrix}$$

The 2-D DCT could then be computed using the tensor product of the corresponding 1-D transform, i.e., using the Seperability property process the 2-D input array by implementing 1-D transform along its rows and columns consecutively as $X = A_N X A'_N$ where x is the input matrix.

A. Computational Complexity

(i) *Multiplications*

The Multiplication operations for the new DCT algorithm could be derived from two sections: the recursive kernel and the transformation matrix T.

The transformation matrix involves computation of cosine values, which require floating-point multiplications. Use of Ramanujan Numbers eliminates this requirement and computes the cosine values through shift and additions only. Thus, for an N length sequence, we need N/2 shifts.

For the N-length sequence, let the number of multiplicative operation with the recursive kernel be represented as M_N. The basic 2X2 kernel involves 2 shift operations. Thus, from equation (13) the number of shifts for $\begin{bmatrix} C_{\frac{N}{2}} & 0 \\ 0 & C_{\frac{N}{2}} \end{bmatrix}$ is $2M_{\frac{N}{2}}$. Hence the total number of shifts are $\frac{N}{2}\log_2 N$.

(ii) *Additions:*

The additive operations could also be grouped into two parts: The recursive kernel and $[P_L]_N$. Let the number of additions from the recursive kernel be represented as A_{1N} and that from the $[P_L]_N$ be A_{2N}.

The transformation matrix involving computation of cosine values using Ramanujan Number requires $N/2$ additions for an N-length sequence.

The basic 2×2 kernel requires 2 additions, hence, from equation (13), the number of additions for the recursive kernel is $2A_{1\frac{N}{2}}$ and that from $[P_2]_N$ is N. So, $A_{1N} = 2A_{1\frac{N}{2}} + N$, with $A_{14} = 8$. By mathematical induction we have, $A_{1N} = N\log_2 N$.

From equation (14), the number of additions is $A_{2N} = 2A_{2\frac{N}{2}} + \frac{N}{2} - 1$, with $A_{24} = 1$. By mathematical induction, we have $A_{2N} = \frac{N}{2}\log_2 N - N + 1$. So, $A_N = A_{1N} + A_{2N} = \frac{3N}{2}\log_2 N - N + 1$.

VI. Simulation Results

To illustrate the efficiency and the rationality of the proposed new algorithm, we compared with the other existing algorithms and the results are tabulated in Table II.

Table II shows that Direct Fast DCT algorithms require, if N=8, 12 floating-point multiplications and 29 additions. The need of floating-point multiplications is eliminated by multiplierless integer transforms which takes 8 integer multiplications, and 35 additions, but requires additional lifting steps which is 15 for N=8.

TABLE II

COMPARISON OF COMPUTATIONAL COMPLEXITY

Operations	Direct Fast DCT [8]	Integer DCT [14]	Proposed Ramanujan DCT
Multiplications	N/2 l og₂ N (floating-point)	N (integer)	N/2 l og₂ N (shifts)
Additions	(3N/2log₂N)-N+1	(2Nlog₂N)-2N+2	(3N/2log₂N)-N+1
Lifting steps	Nil	(3N/2log₂N)-3N+3	Nil

Our proposed algorithm eliminated the necessity of multipliers completely by using shifts since the numbers could be represented as powers of 2. For N=8, Ramanujan Number DCT requires 12 shifts, and 29 additions with no additional requirement of the lifting steps.

VII. Conclusion

This paper discusses the novel technique of computing 2-D DCT in the recursive structure using the Ramanujan Numbers. If N, which represents the Block size, is a Ramanujan Number, then the proposed algorithm computes DCT Coefficients using $N/2\log_2 N$ shifts and $(3N/2\log_2 N)-N+1$ additions which is less than the multiplierless integer transform used in H.264/AVC standard. Thus, the proposed fast algorithm can achieve the higher speed in real-time H.264/AVC video signal processing.

References

[1] K.R.Rao, P.Yip, "Discreate Cosine Transform Algorithms, Advantages Applications". New York: Academic 1990.
[2] H.S. Hou, "A Fast Recursive Algorithms for Computing the Discrete Cosine Transform". IEEE Trans. Acoust., Speech, Signal Processing, Vol.35, pp 1455-1461, Oct 1987.
[3] N.I.Cho, S.U.Lee, "A Fast 4X4 DCT Algorithm for the Recursive 2-D DCT". IEEE Trans. Signal Processing, Vol.40, pp 2166-2173. Sep 1992.
[4] C.W.Kok, "Faast Algorithm for Computing Discrete Cosine Transforms". IEEE Trans. Signal Processing, Vol.45, No.3, pp 757-760. March 1977.

[5] T.Tran,"The BinDCT: Fast multiplier less approximation of the DCT". IEEE Signal Proc.141-14 (2000).
[6] Yonghong Zeng, Lizhi Cheng, Guoan Bi, and Alex C. Kot, "Integer DCT's and Fast Algorithms".IEEE Transactions on Signal Processing Vol 49, No.11, November 2001.
[7] Xu Hui, Cheng Lizhi,"An Integer Hierarchy:Lapped Biorthogonal Transform via Lifting Steps and Application in Image Coding". Proc. ICSP-02, 2002.
[8] Z.D.Wang, "Reconsideration of 'A Fast Computational Algorithm for the Discrete Cosine Transform'," . IEEE Trans. Comm., Vol.COM:31, pp121-123. Jan 1983.
[9] W.H. Chen, C.H. Smith, and S.C.Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform". IEEE Trans. Comm. Vol.COM:25, No.9, pp 1004-1009. Sep 1997.
[10] Nirdosh Bhatnagar, "On computation of certain discrete Fourier transforms using binary calculus".Signal Processing-Elsevier Vol 43,1995
[11] Nirdosh Bhatnagar, "DFT computation using shift and addition operations".IEEE Int.Conf. on ASSP, Georgia, 1996.
[12] V.K. Ananthashayana, K. Chidanandagowda, "Fast Discrete Cosine Transform using Ramanujan Numbers and its application to image compression", Int. Conf. On Combinatories, Statistics, pattern recognition and related areas, University of Mysore, Mysore, India, pp.30, Dec. 28-30, 1998.
[13] Geetha.K.S, V.K.Ananthashayana,"Fast Multiplierless Recursive transforms using Ramanujan Numbers".Proc. IEEE International Conference on Multimedia, Signal Processing and Communication Technologies, Aligarh, India. March 2008.
[14] Y.Zeng, L.Cheng,"Integer DCTs and Fast Algorithms"., IEEE Trans. on Signal Processing, Vol.49,No.11, November 2001.
[15] Chih-Peng, Fan, "Fast 2-Dimensional 4X4 Forward Integer Transform Implementation for H.264/AVC". IEEE Trans. on Circuits and systems. Vol.53, Issue 3, pp174-177. March 2006.
[16] Ji Xiuhua, Z. Caiming, W.Yanling, "Fast Algorithm of the 2-D 4X4 Inverse Integer Transform for H.264/AVC". 0-7695-2882-1/07, IEEE 2007.