A Novel Cosine Approximation for High-Speed Evaluation of DCT

Geetha K.S geethakomandur@gmail.com
Assistant Professor, Dept of E&CE
R.V.College of Engineering,
Bangalore-59, India

M.UttaraKumari dr.uttarakumari@gmail.com
Professor, Dean of P.G.Studies
Dept of E&CE
R.V.College of Engineering,
Bangalore-59, India

Abstract

This article presents a novel cosine approximation for high-speed evaluation of DCT (Discrete Cosine Transform) using Ramanujan Ordered Numbers. The proposed method uses the Ramanujan ordered number to convert the angles of the cosine function to integers. Evaluation of these angles is by using a 4th degree polynomial that approximates the cosine function with error of approximation in the order of 10^{-3}. The evaluation of the cosine function is explained through the computation of the DCT coefficients. High-speed evaluation at the algorithmic level is measured in terms of the computational complexity of the algorithm. The proposed algorithm of cosine approximation increases the overhead on the number of adders by 13.6%. This algorithm avoids floating-point multipliers and requires $N/2\log_2 N$ shifts and (3N/2 log2 N)- N + 1 addition operations to evaluate an N-point DCT coefficients thereby improving the speed of computation of the coefficients

Keywords: Cosine Approximation, High-Speed Evaluation, DCT, Ramanujan Ordered Number.

1. INTRODUCTION

High-speed approximation to the cosine functions are often used in digital signal and image processing or in digital control. With the ever increasing complexity of processing systems and the increasing demands on the data rates and the quality of service, efficient calculation of the cosine function with a high degree of accuracy is vital. Several methods have been proposed to evaluate these functions [1]. When the input/output precision is relatively low(less than 24 bits), table and addition methods are often employed [2, 3]. Efficient methods on small multipliers and tables have been proposed in [4]. Method based on the small look up table and low-degree polynomial approximations with sparse coefficients are discussed in [5, 6].

Recently, there has been increasing interest in approximating a given floating-point transform using only very large scale integration-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. Normally the multiplierless approximation are discussed for implementing the discrete cosine transform (DCT) which is widely used in image/video coding applications. The fast bi-orthogonal Binary DCT (BinDCT) [7] and Integer DCT (IntDCT) [8, 9] belong to a class of multiplierless transforms which compute the coefficients of the form ωn. 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.

In this work, we show that using Ramanujan ordered numbers; it is possible to evaluate the cosine functions using only shifts and addition operations. The complete operator avoids the use of floating-point multiplication used for evaluation of DCT coefficients, thus making the algorithm completely multiplierless. 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.

2. RAMANUJAN ORDERED NUMBERS

Ramanujan ordered Numbers are related to π and integers which are powers of 2. Ramanujan ordered Number of degree-1 was used in [10,11] to compute the Discrete Fourier Transform. The accuracy of the transform can be further improved by using the Ramanujan ordered number of degree-2. This is more evident in terms of the errors involved in the approximation.

2.1 Definition : Ramanujan ordered Number of degree-1

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

$$\Re_1(a) = \left[\frac{2\pi}{l_1(a)}\right] \quad where\ l_1(a) = 2^{-a} \tag{1}$$

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 $l_1(2) = 2^{-2}$,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.

(a)	$\Re(a)$	$\hat{\pi}$	Upper bound of error
0	6	3.0	4.507×10^{-2}
1	13	3.25	3.4507×10^{-3}
3	50	3.125	5.287×10^{-3}

TABLE 1: Ramanujan ordered Number of Degree-1.

Table 1 clearly shows the numbers which can be represented as Ramanujan ordered -numbers of order-1. Normally, the digital signal processing applications requires the numbers to be power of 2; hence higher degree numbers are required for all practical applications.

2.2 Definition: Ramanujan ordered Number of degree-2

The Ramanujan ordered Number of degree-2 [11] 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 degree-2 are,

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

$$\iota_{21}(l,m)=2^{-l}+2^{-m} \quad for\ m>l\geq 0$$
$$\iota_{22}(l,m)=2^{-l}-2^{-m} \quad for\ (m-1)>l\geq 0 \qquad (4)$$

Where l and m are integers, Hence

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

Ramanujan ordered Number of degree-2 and their properties are listed in the table 2 below.

(l,m)	$\Re(l,m)$	$\hat{\pi}$	Upper bound of error
0,2	5	3.125	5.28x10^{-3}
1,2	8	3.0	4.507x10^{-2}
4,5	67	3.140	5.067x10^{-4}

TABLE 2: Ramanujan ordered Number of Degree-2.

The accuracy of the numbers increase with the increase in the degree of the Ramanujan ordered numbers at the expense of additional shifts and additions. State-of-art technologies in Image processing uses the block processing techniques for applications like image compression or image enhancement. The standardized image block size is 8×8, which provides us an opportunity to use Ramanujan ordered numbers to reduce the complexity of the algorithms. Table III shows the higher degree Ramanujan ordered Numbers and their accuracies.

$(l,m,p.....)$	$\Re(l,m,p....)$	$\hat{\pi}$	Upper bound of error	Computational complexity	
				Shifts	Adds
1,2	8	3. 0	4.507x10^{-2}	2	1
1,2,5	8	3.125	5.28x10^{-3}	3	2
1,2,5,8	8	3.1406	3.08x10^{-4}	4	3

TABLE 3: Ramanujan ordered Number of Higher Degree.

Table 3 shows that the error of approximation decreases with the increase in the degree of Ramanujan ordered numbers, but the computational complexity also increases. Hence the choice of Ramanujan ordered number of degree-2 is best validated for the accuracy and the computational overhead in computation of the cosine functions.

3. COSINE APPROXIMATION USING RAMANUJAN ORDERED NUMBER

Method of computing the cosine function is to find a polynomial that allows us to approximate the cosine function locally. A polynomial is a good approximation to the function near some point, $x = a$, if the derivatives of the polynomial at the point are equal to the derivatives of the cosine curve at that point. Thus the higher the degree of the polynomial, the better is the approximation. If p_n denotes the nth polynomial about $x = a$ for a function f, and if we approximate $f(x)$ by $p(x)$ at a point x, then the difference $R_n(x) = f(x) - p(x)$ is called the nth remainder for f about $x = a$. The plot of the 4th order approximation along with the difference plot is as shown in figure 1.

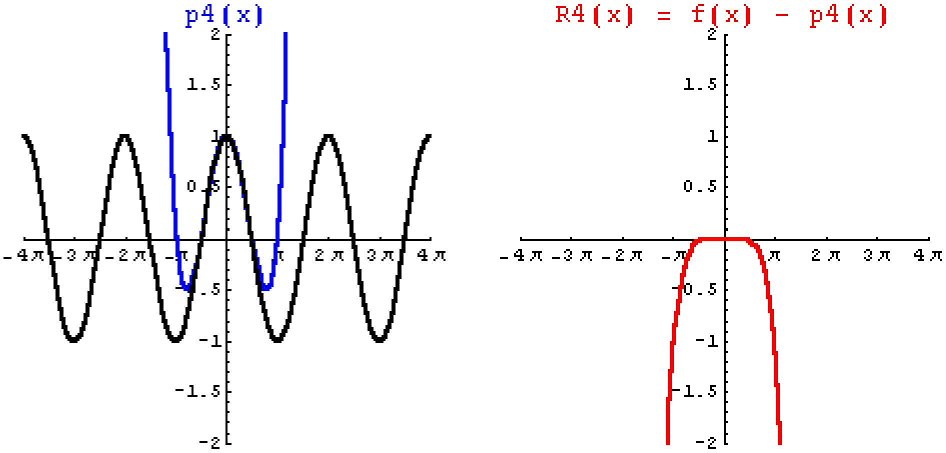


FIGURE 1: Plot of f(x)=cos(x) and the remainder function .

Figure 2 indicates that the cosine approximation at various angles with 4th degree polynomial is almost close with the actual values. The accuracy obtained at various degrees of the polynomial is compared in table 4. Thus we choose the 4th degree polynomial for the cosine approximation.

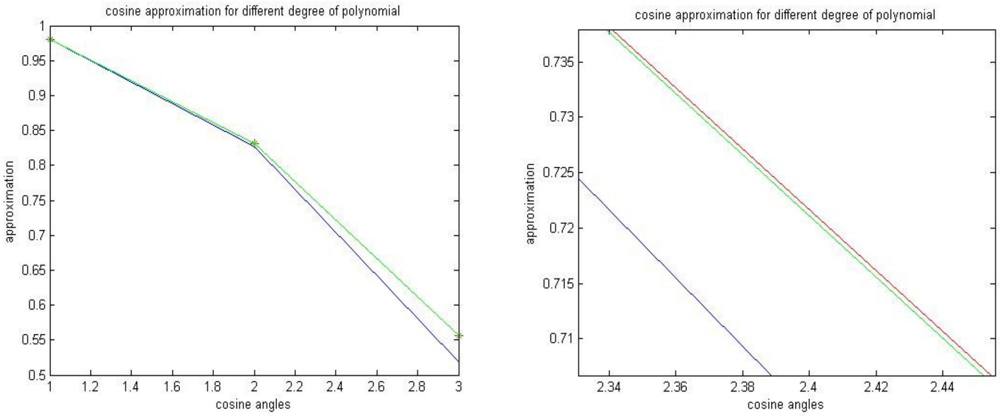


FIGURE 2: Expanded version of cosine approximation.

$\cos(5\pi/16)$	[illegible]	[illegible]	[illegible]

TABLE 4: Cosine approximation in comparison with the Actual value

The evaluation of the cosine function using the polynomial approximation is explained through the application of computing the Discrete Cosine Transform coefficients which uses the cosine as its basis function. Discrete Cosine Transforms (DCT) is widely used in the area of signal processing, particularly for transform coding of images. Computation of DCT coefficients involves evaluation of cosine angles of multiples of 2π/N.

The input sequence $\{x_n\}$ of size N, is transformed as, $\{y_k\}$. The transformation may be either DFT or DCT. The DCT defined as [12]

$$y_k = \frac{2\varepsilon_k}{N}\sum_{n=0}^{N-1} x_n \cos\left(\pi k \frac{(2n+1)}{2N}\right)$$
$$for\ k = 0,1...N-1$$
$$\varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & for\ \ k = 0 \\ 1 & otherwise \end{cases} \tag{5}$$

Neglecting the scaling factors, the DCT kernel could be simplified as

$$c_n = \cos\left(\frac{2\pi}{N} 2^{-2}(2n+1)k\right)$$
$$for\ 0 \le n \le N-1,\ 0 \le k \le N-1 \tag{6}$$

DCT coefficients are computed by evaluating the sequences of type

$$\{c_n / c_n = p\cos(2\pi n / N), n = 0,1,2...(N-1), p \in \mathrm{R}\} \tag{7}$$

where R is the set of real numbers. These computations are done via a Chebyshev-type of recursion. Let us define

$$\mathrm{w}(M,p) = \{w_n / w_n = p\cos(2\pi n/M)\}$$
$$n = 0,1,2,....\Psi, p \in \mathrm{R} \tag{8}$$

$$\Psi = \left(\frac{M}{4} - 1\right), M = \beta N \qquad (9)$$

$$\beta = \begin{cases} 1 & if\ \ 4\ divides\ N \\ 2 & if\ \ 2\ divides\ N\ but\ not\ 4 \\ 4 & otherwise \end{cases} \qquad (10)$$

The use of β facilitates the computation of W (M, p) by considering cosine values from the first quadrant of the circle. Then the sequence {c$_n$} can be evaluating recursively$\{W_n\}$.

The W (M, p) sequence is estimated as follows. Let us define $\frac{2\pi}{N}2^{-2}$ as x and hence $c_n = \cos(nx)$. We approximate x by $\hat{x}$ equal to $2^{-l} + 2^{-m}$ with l and m being non-negative integers. Thus, we approximate c_n's by t_n's, where α is equal to $\hat{x}^2/2 - \hat{x}^4/4!$. Since $\hat{x}$ is a Ramanujan ordered Number of degree-2, α is of the form $2^{-c} + 2^{-d}$, where c and d are integers. Then

$$\begin{aligned} & t_0 = p \\ & t_1 = p(1-\alpha) \\ & \vdots \\ & t_{n+1} = 2(1-\alpha)t_n - t_{n-1} \\ & n = 1, 2, \cdots, (\Psi - 1) \end{aligned} \qquad (11)$$

The above equations show that the recursive equations can be computed by simple shift and addition operations.

4. SIMULATION RESULTS

To evaluate the performance of the proposed cosine approximation, the following cosine angles were simulated in matlab. The actual value is the angles evaluated using the inbuilt COS function and the values obtained from the proposed approximation are tabulated in Table 5. The deviation of the approximated results from the actual value is tabulated as error in table 5. The error due to approximation is of the order of 10^{-3} which is acceptable for image coding applications.

Function	Actual value	Proposed Approximation	Error
$\cos(\pi/16)$	0.980785	0.9810	-2.15×10^{-4}
$\cos(3\pi/16)$	0.8314696	0.83312	-1.6504×10^{-3}
$\cos(5\pi/16)$	0.55557	0.5571	-1.53×10^{-3}

TABLE 5: Comparison of the proposed cosine approximation

The cosine approximation being multiplierless using Ramanujan ordered Number proved to be more efficient and simple when being applied to image coding. We tested the efficacy of the proposed algorithm by replacing the existing floating-point DCT/IDCT block from the JPEG encoder/decoder. To prove the efficiency of the proposed algorithm standard test images like

Cameraman and Lena images were considered for JPEG encoder/decoder. The floating-point DCT block evaluates 2-D DCT on the 8x8 block of the image by implementing the direct function as given in equation (5). Most commonly used algorithms [13, 14] use lookup tables to access the cosine values to compute the DCT coefficients and hence require floating-point multipliers in hardware implementation of the algorithm. Table 6 shows the comparison of the algorithms in terms of number of adders and table 7 gives the comparison in terms of shifters/multipliers required for the hardware implementation. The computational complexity is reduced by completely avoiding the floating-point multiplications and is replaced by N/2log$_2$N shift operations. However the number of addition operations required to compute N DCT coefficients is (3N/2 log2 N)- N + 1 which increases by 13.6% when compared with the number of adders required for the floating-point DCT.

Transform	Floating-point DCT [13]	Proposed Ramanujan DCT
8 x 8	29	36
16 x 16	81	96
32 x 32	209	240
64 x 64	513	576

TABLE 6: Computational Complexity in terms of additions

Transform	Floating-point DCT [13] (Floating-point multiplications)	Proposed Ramanujan DCT (Shifts)
8 x 8	12	12
16 x 16	36	36
32 x 32	80	80
64 x 64	384	384

TABLE 7: Computational Complexity in terms of Multiplications

The performance of the proposed algorithm is compared with the existing floating-point DCT in terms of PSNR and the Compression Ratio achieved and the results are tabulated in Table 8.
The PSNR is computed using the MSE as the error metric.

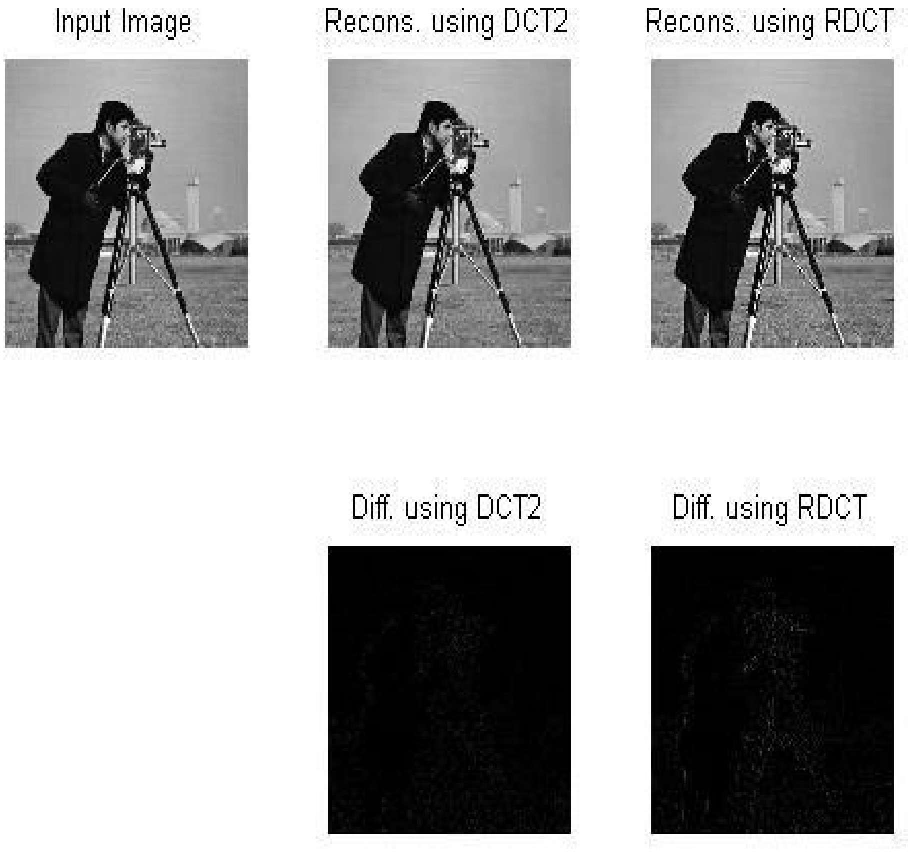
$$MSE = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[I(m,n) - I^{'}(m,n)\right]^2$$

$$PSNR = 20 * \log_{10}\left(255/\sqrt{MSE}\right) \tag{12}$$

Compression Ratio =Uncompressed Image size / Compressed image size.
Where N x M is the size of the image, I(m,n) is the original image and I`(m,n) is the reconstructed image.
The results show that the reduction in PSNR for color images like Football is around 10% and reduction in PSNR for smooth transition images like Lena, Cameraman is around 4%, and is improved by 4% for sharp transition images like Circuit board images.

Image	Proposed Ramanujan DCT		Floating-point DCT[13]	
	Compression Ratio	PSNR(dB)	Compression Ratio	PSNR(dB)
Circuit	5.30:1	72.93	5.6:1	69.01
Football	5.62:1	69.24	5.97:1	77.20
Lena	4.20:1	62.04	4.56:1	65.77
Medical Image	5.74:1	62.91	5.74:1	68. 1

TABLE 8: Performance comparison of Ramanujan DCT and Floating-point DCT



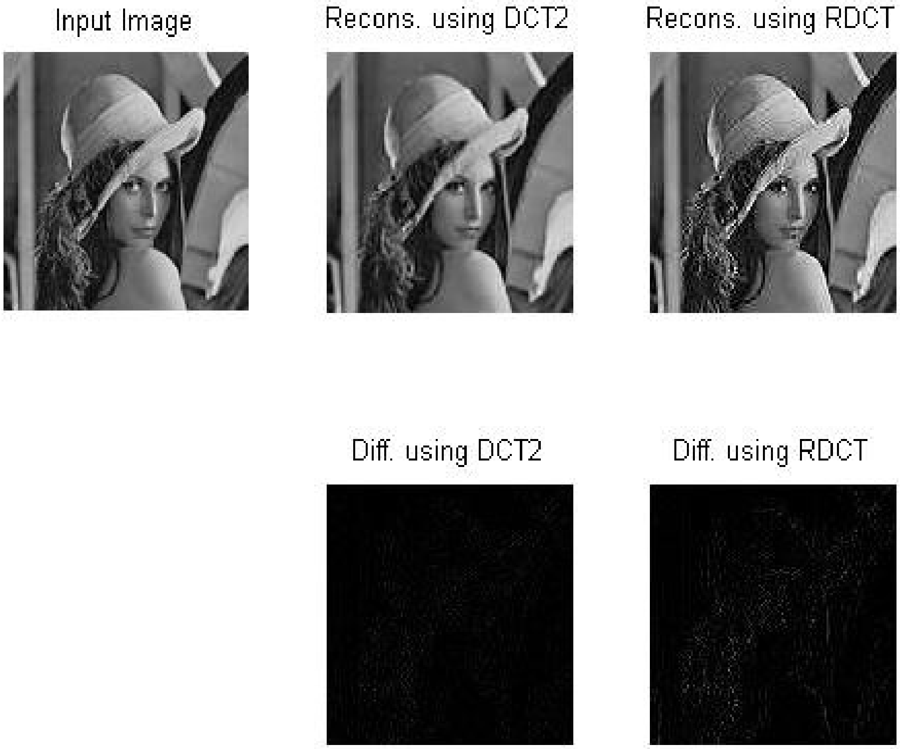


FIGURE3: The Original image and Reconstructed image using floating-point DCT and Ramanujan DCT.

Figure 3 shows the original image and the reconstructed image obtained using both RDCT and DCT2 function applied in JPEG compression standard technique. The difference image obtained between the original and reconstructed image shows that the error in cosine approximation is very negligible.

5. CONCLUSIONS

We have presented a method for approximation of the cosine function using Ramanujan ordered number of degree 2. The cosine function is evaluated using a 4th degree polynomial with an error of approximation in the order of 10^{-3} . This method allows us to evaluate the cosine function using only integers which are powers of 2 thereby replaces the complex floating-point multiplications by shifters & adders. This algorithm takes N/2 $\log_2$ N shifts and (3N/2 log2 N) - N + 1 addition operations to evaluate an N-point DCT coefficients. The cosine approximation increases the overhead on the number of adders by 13.6%. The proposed algorithm reduces the computational complexity and hence improves the speed of evaluation of the DCT coefficients. The proposed algorithm reduces the complexity in hardware implementation using FPGA. The results show that the reconstructed image is almost same as obtained by evaluating the floating-point DCT.

6. REFERENCES

1. J.-M.Muller, Elementary Functions: Algorithms and Implementation, Birkhauser, Boston, 1997.

2. F.de Dinechin and A.Tisserand, “Multipartite table methods,” IEEE Transactions on Computers, vol.54, no.3,pp. 319-330, Mar. 2005.

3. M.Schulte and J. Stine, “Approximating elementary functions with symmetric bipartite tables,” IEEE Transactions on Computers, vol. 48, no. 8, pp. 842-847, Aug. 1999.

4. J. Detrey and F.de Dinechin, “Second order function approximation using a single multiplication on FPGAs,” in 14th International Conference on Field-Programmable Logic and Applications, Aug.2004, pp.221-230, LNCS 3203.

5. Arnad Tisserand, “Hardware operator for simultaneous sine and cosine evaluation,” in ICASSP 2006, vol III, pp.992-995.

6. N.Brisebarre, J.-M.Muller, A.Tisserand and S.Torres, “Hardware operators for function evaluation using sparse-coefficient polynomials,” in Electronic Letters 25, 42(2006), pp. 1441-1442.

7. T Tran, “The BinDCT: Fast multiplier less approximation of the DCT”. IEEE Signal Proc. Vol. 7, No. 6, pp. 141-44, 2000.

8. Y Zeng, L Cheng, G Bi, and A C Kot, “Integer DCT’s and Fast Algorithms”. IEEE Trans Signal Processing, Vol. 49, No. 11, Nov. 2001.

9. X Hui and L Cheng, “An Integer Hierarchy: Lapped Biorthogonal Transform via Lifting Steps and Application in Image Coding”. Proc. ICSP-02, 2002.

10. Nirdosh Bhatnagar, “On computation of certain discrete Fourier transforms using binary calculus”.Signal Processing-Elsevier Vol43,1995.

11. Geetha.K.S, M.Uttarakumari, “Multiplierless Recursive algorithm using Ramanujan ordered Numbers,” in IETE Journal of Research, vol. 56, Issue 4, JUL-AUG 2010.

12. K.R.Rao, P.Yip, “Discrete Cosine Transform Algorithms, Advantages Applications”. New York: Academic 1990.

13. 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.

14. 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.