An Efficient Multiplierless Transform Algorithm for Video Coding

Geetha. K. S geethakomandur@gmail.com
Associate Professor, Dept of E&CE
R.V.College of Engineering,
Bangalore, 560 059, India

M K Pushpa pushpachandan@rediffmail.com
Associate Professor, Dept of IT
M.S.Ramaiah Institute of Technology,
Bangalore,560 054, India

Dr.M.Uttarakumari dr.uttarakumari@gmail.com
Professor, Dept of E&C
R.V.College of Engineering,
Bangalore,560 059, India

Dr. S.Sethu Selvi selvi@msrit.edu
Professor & Head, Dept of E&C
M.S.Ramaiah Institute of Technology,
Bangalore,560 054, India

Abstract

This paper presents an efficient algorithm to accelerate software video encoders/decoders by reducing the number of arithmetic operations for Discrete Cosine Transform (DCT). A multiplierless Ramanujan Ordered Number DCT (RDCT) is presented which computes the coefficients using shifts and addition operations only. The reduction in computational complexity has improved the performance of the video codec by almost 58% compared with the commonly used integer DCT. The results show that significant computation reduction can be achieved with negligible average peak signal-to-noise ratio (PSNR) degradation. The average structural similarity index matrix (SSIM) also ensures that the degradation due to the approximation is minimal.

Keywords: Ramanujan Ordered Number DCT, Multiplierless DCT, Video Coding.

1. INTRODUCTION

Digital video applications have become more and more popular in our everyday life. Currently, there are several video standards, such as H.261 [I], H.263 [2], and MPEG [3][4], established for different applications. All these standards use motion compensated prediction, Discrete Cosine Transform (DCT), quantization, zigzag scan, and Variable Length Coding (VLC) as their basic functional blocks. Among these building blocks, Motion estimation (ME) in the motion compensated (MC) prediction is the most computationally intensive part, and then the DCT and the Inverse DCT (IDCT). Many fast algorithms have been developed to speed up the computation for Motion estimation. In this paper, an efficient technique is investigated to accelerate software video encoders by reducing the number of operations for DCT and quantization. The DCT and the quantization processes require a lot of multiplications, which are computationally expensive. A modification is proposed by replacing the 2-D DCT block in the standard MPEG-2 video codec with the 2-D Multiplierless Recursive DCT block. The performance is then compared with the existing DCT algorithms.

The organization of the paper is as follows. In Section 2, different blocks of video coding as in MPEG coder/decoder are explained, Section 3, explains the use of the multiplierless DCT coefficient computation that reduces the computation in the video encoder. In Section 4, the methodology of the proposed technique with the simulation results is discussed.

2. MPEG CODER/DECODER

The international standard [5, 6] describe a system, MPEG-2, for encoding and decoding digital video data. The standard allows for the encoding of video over a wide range of resolutions, including higher resolutions commonly known as HDTV.

In this system, encoded pictures are made up of pixels. If each 8×8 array of pixels is known as a block, then an 2×2 array of blocks is termed a macroblock. In this paper, an 8×8 array of pixels is used as macroblock. Compression is achieved using the well known techniques of prediction (motion estimation in the encoder, motion compensation in the decoder), 2-D DCT, quantization of DCT coefficients, and Huffman/run(remove space) length coding. Pictures called I pictures are encoded without prediction and maintained as reference frames. Pictures termed P pictures may be encoded with prediction from previous pictures. B pictures may be encoded using prediction from both previous and subsequent pictures. A simplified MPEG-2 encoder and decoder is shown in Figure 1.

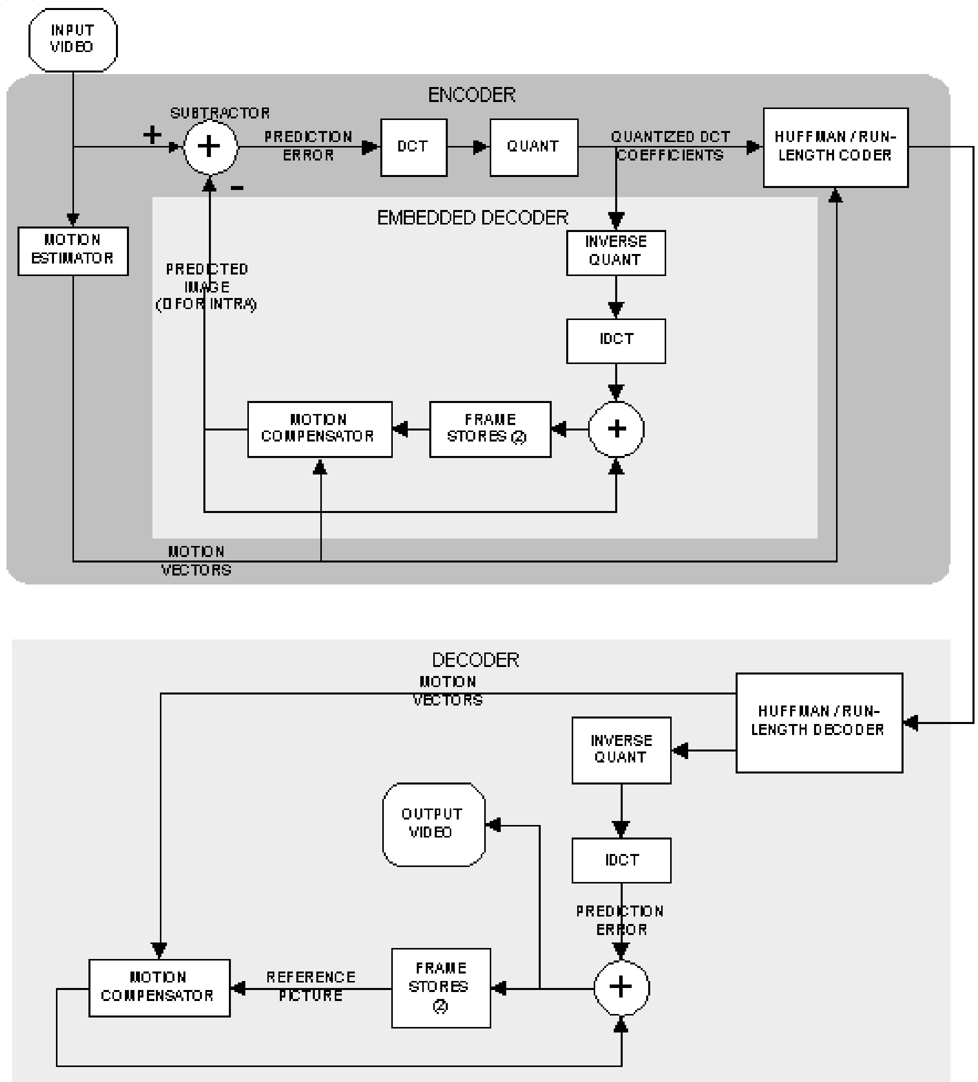

FIGURE1. MPEG-2 Encoder and Decoder

Before DCT is performed, motion compensated prediction is done for every macro block (8×8 pixels) on inter-coded frames. The objective of motion estimation is to find the best match of the current macro block within the search region in the reference frame. The common matching criterion used for finding the best match in the search region is the Mean Absolute Difference (MAD).

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left| C_{ij} - R_{ij} \right| \quad (1)$$

Where N is the size of the macro block, C_{ij} and R_{ij} are the pixels being compared in current macro block and the reference macro block respectively.

In motion compensated predictive coding, before performing the DCT computation, the Three Step Search algorithm [7, 8] is used to find the motion vectors. The best macroblock is found by using the MAD as a measure. The search algorithm is started with the search location at the centre of the macroblock as (0, 0). The step size is then fixed as S=4, and the search parameter as 7 for a macroblock of size 8×8. So, the search continues for the eight neighborhood pixels around location (0, 0). Out of these 9 locations, the pixel with the least cost function is then reinitiated as the new search origin and the step size is then reduced by half. So, S=S/2. The procedure is repeated until S=1. The pixel with the least cost function would then be the best match. The vector that represents the best match is saved.

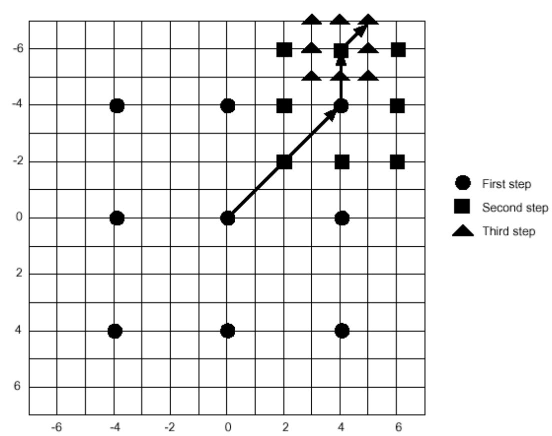

FIGURE 2: Three Step Search Procedure. (Motion Vector is (5,7)

Each motion compensated macro block consists of four 8×8 luminance and two 8×8 chrominance prediction error blocks (difference blocks). These 8×8 blocks are transformed to generate 8×8 DCT coefficients and these coefficients are quantized for compression.

3. PROPOSED VIDEO CODEC

The DCT and the quantization processes require a lot of multiplications, which are computationally expensive. The standardized DCT block requires floating-point multipliers and for an 8×8 block, evaluation of coefficients require 12 floating-point multipliers. The implementation of such a codec is more expensive as the complexity is concentrated towards the floating-point multipliers. This disadvantage is overcome by replacing the floating-point DCT block with a multiplierless DCT block where the coefficients are evaluated using Ramanujan ordered numbers. Computation of DCT coefficients involves evaluation of cosine angles of multiples of 2π/N. Evaluation of these angles is accomplished by using a 4^{th} degree polynomial that approximates the cosine function with error of approximation in the order of 10^{-3} [13] . 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 ordered numbers. Use of Ramanujan ordered Number for computing DCT was outlined by the author in [11,12]. Matrix factorization of the transformation matrix reduced the complexity to $\frac{N}{2}\log_2 N$ shifts and $\frac{3}{2}N\log_2 N - N + 1$ additions [12] thereby eliminating the use of multipliers.

3.1 Multiplierless Ramanujan Ordered Number DCT(RDCT)[11,12]

The 2-D Discrete Cosine Transform (DCT) can be defined as follows:

$$C(k_1,k_2)=\frac{4}{N_1N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x(n_1,n_2)\cos\left(\pi k_1\frac{(2n_1+1)}{2N_1}\right)\cos\left(\pi k_2\frac{(2n_2+1)}{2N_2}\right) \qquad (2)$$

Neglecting the scaling factors and using the property of Seperability, the DCT equation can be written as:

$$C(k_1,k_2)=\sum_{n_1=0}^{N_1-1}\left(\sum_{n_2=0}^{N_2-1}x(n_1,n_2)\cos\left(\pi k_2\frac{(2n_2+1)}{2N_2}\right)\right)\cos\left(\pi k_1\frac{(2n_1+1)}{2N_1}\right) \qquad (3)$$

Thus, 2-D $N_1 \times N_2$ DCT can be implemented by computing the row transformation followed by the column transformation. Hence, a 1-D transformation can be considered as a process of evaluating the sequences in the form as follows:

$$c_n=\sum_{n=0}^{N-1}x(n)\cos\left(\frac{2\pi}{N}2^{-2}(2n+1)k\right) \qquad (4)$$

3.1.1 Evaluation of Transform Coefficients Using Chebyshev Recursion

Computation of DCT coefficients requires evaluation of sequences of type

$$\left\{c_n \mid c_n = p\cos\left(2\pi n/N\right) n=0,1,2\ldots N-1,\ p\in\Re\right\} \qquad (5)$$

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

$$\begin{aligned} W(M,p) &= \left\{w_n \mid w_n = pcos\left(2\pi n/M\right)\right\} \\ n &= 0,1......\Psi, \quad p\in\Re \qquad (6) \\ \Psi &= \left(\frac{M}{4}-1\right), \quad M=\beta N \end{aligned}$$

where, β is equal to 1, if N is divisible by 4. It is equal to 2, if N is divisible by 2, but not by 4. Otherwise, it is equal to 4(N is not divisible by 2). The use of β facilitates the computation of w (*M, p*) by considering cosine values from the first quadrant of the circle.

Let us then define

$$x = \frac{2\pi}{N} 2^{-2}$$
$$\therefore\ w_n = \cos\left(\left(2n+1\right)x\right) \quad (7)$$

x is then represented using Ramanujan ordered number of degree 2 as $\hat{x} = 2^{-l} + 2^{-m}$ *where l and m are non-negative integers*.

For ex: If N=8, then

$$x = \frac{2\pi}{8} 2^{-2} \cong \left(2^{-1} + 2^{-2}\right) 2^{-2}$$
$$\hat{x} = 2^{-3} + 2^{-4} \quad (8)$$
$$\therefore\ w_n = \cos\left(n'\hat{x}\right)$$

where n' is the scaled and shifted time samples and $\hat{x}$ being the Ramanujan ordered number. Evaluation of these cosine values is by cosine approximation using 2nd order polynomial. Let the polynomial be defined as

$$\alpha = \frac{\hat{x}^2}{2!}$$
$$\therefore\ t_n\left(\alpha\right) = \cos\left(n\alpha\right) \quad (9)$$

$t_n\left(\alpha\right)$ are then computed using the recursive equations as

$$t_0\left(\alpha\right) = 1$$
$$t_1\left(\alpha\right) = \left(1-\alpha\right)$$
$$\vdots$$
$$t_{n+1}\left(\alpha\right) = 2\left(1-\alpha\right)t_n\left(\alpha\right) - t_{n-1}\left(\alpha\right)$$
$$n = 1, 2....., \left(\Psi - 1\right) \quad (10)$$

It is observed that the above recursive equations are closely related to Chebyshev polynomial of the first kind. Since the evaluation of the recursive equations involve only numbers of powers of two, $t_n\left(\alpha\right)$'s and therefore $c_n\left(\alpha\right)$'s can be computed by simple shift and addition operations. RDCT kernel needs samples only at $\left(2n+1\right)$, and thus all the samples of $t_n\left(\alpha\right)$ need not be stored.

TABLE I. COMPARISON OF COMPUTATIONAL COMPLEXITY

Operations	Floating-point DCT $N \times M$ [9]	Integer DCT $N \times M$ [10]	RDCT $N \times M$ [11]
Multiplications	$\left(NM/2\right)\log_2 M$ (Floating-point)	NM (Integer)	Nil
Additions	$\left(3NM/2\right)\log_2 M$ $+\left(-2NM + N + M\right)$	$2\left(\log_2 NM - 1\right)$ $+NM+2$	$\left(3NM/2\right)\log_2 M$ $-2NM + N + M$
Lifting Steps	Nil	$\left(3N/2\right)\log_2 N$ $-3N+3$	Nil
Shifts	Nil	Nil	$\left(NM/2\right)\log_2 M$

Table I gives the comparison of the reduction of the computational complexity of the proposed algorithm. To compute $N \times M$ DCT the proposed algorithm takes $(3NM/2)\log_2 M - 2NM + N + M$ additions and $NM/2\log_2 M$ shift operations. Thus for N=M, the proposed algorithm for a 8×8 block DCT evaluation, requires 96 shift operation, and 176 addition operations. The proposed algorithm being recursive ensures that the storage of the trigonometric values is not required.

4. SIMULATION RESULTS

To demonstrate the efficacy of the proposed algorithm on MPEG Video codec, the results were compared with the existing algorithm of the standard MPEG-2 video codec and the results are tabulated. The proposed RDCT is tested by replacing the two-dimensional DCT block in the MPEG-2 standard algorithm with the 2-D RDCT block. The performance is then compared by using commonly used multiplierless 2-D Integer DCT. --. DC coefficient is quantized and coded separately and transmitted. The AC coefficients are encoded with very few coefficients removing the completely zero coefficients block.

Table.II gives the average PSNR of the original frame with decoded frame, using 60 frames of input video sequence (video grabbed at 30fps), with a GOP (group of pictures) as 10 and the encoding format as $I_1P_4B_2B_3P_7B_5B_6I_{10}B_7B_8$. The step size is considered as 10 to decode all 10 frames in the display format as $I_1B_2B_3P_4B_5B_6P_7B_8B_9I_{10}$. The simulation has been evaluated for both forward and bidirectional prediction and the results shows that the motion estimation in both the formats gives better results for the proposed RDCT when compared with the floating-point DCT and the integer DCT. From Table II it is clear that the proposed RDCT offers same accuracy in average PSNR as that of the floating-point DCT with a deviation of 0.01%, and the deviation with Integer DCT is by 0.01% for standard test sequence like Alex.avi. The deviation in PSNR of the RDCT with floating-point DCT is 0.005% and with integer DCT is 0.08% for real time data sequence. This clearly shows that the proposed technique of using RDCT for the video codec is providing better reconstructed picture quality.

TABLE II AVERAGE PSNR IN dB OF THE DECODED FRAMES

Test Sequence	Frame Format	Floating-point DCT[9]	Integer DCT [10]	Multiplierless RDCT
Real time Data (Frames grabbed at 30 fps)	IBBPBBPBBP	35.7010	35.6694	35.6991
	IPPPPPPPPP	33.6581	33.6132	33.6525
San_Fran_Traffic	IBBPBBPBBP	34.8076	34.7776	34.7996
	IPPPPPPPPP	31.7476	31.7176	31.7462
Alex	IBBPBBPBBP	36.0928	36.0809	36.0876
	IPPPPPPPPP	31.583	31.5756	31.579

The Structural Similarity Index Matrix (SSIM) index seeks to separately discover differences in local image luminance l(x,y), contrast c(x,y) and structure s(x,y) between the original and compensated images. Given the pixel points (x,y), the SSIM is defined as

$$SSIM(x,y) = l(x,y).c(x,y).S(x,y)$$
$$= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1^2} \cdot \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2^2} \cdot \frac{\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C_3} \tag{11}$$

where μ_x, μ_y, σ_x, σ_y and σ_{xy} are the local sample means, variances, and cross-covariance of x and y. The constants C_1, C_2, C_3 stabilize SSIM when the means and variances become small.

SSIM index varies between 0(worst) and 1(best). Table III shows the average SSIM for decoded frames with original frames.

Table III AVERAGE SSIM BETWEEN THE DECODED AND ORIGINAL FRAMES

Test Sequence	Frame Format	Floating-point DCT[9]	Integer DCT [10]	Multiplierless RDCT
Real time Data (Frames grabbed at 30 fps)	IBBPBBPBBP	0.9223	0.9218	0.9223
	IPPPPPPPPP	0.921645	0.921628	0.921635
San_Fran_Traffic	IBBPBBPBBP	0.85028	0.85020	0.85026
	IPPPPPPPPP	0.85701	0.85014	0.85693
Alex	IBBPBBPBBP	0.8689	0.8684	0.8690
	IPPPPPPPPP	0.8678	0.8668	0.8682

From Table III, it is clear that the quality of decoding is very good with RDCT and achieves the same performance as that of the floating-point DCT. This is ensured by taking the difference frame between the reference frame and the decoded frame. The difference frame is as shown in the Figure 3a and 3b. The difference between the RDCT and the floating-point DCT in terms of SSIM is 0.01% for standard test sequence like Alex.avi and the difference between the RDCT and the Integer DCT in terms of SSIM is 0.07% for the same test sequence. For the real time data the difference between RDCT and floating-point DCT is 0 in terms of SSIM and between RDCT and integer DCT is 0.05% in terms of SSIM. These values clearly indicate that the reconstructed frame with proposed RDCT is very good in subjective quality when compared with the reconstructed frame with Integer DCT.

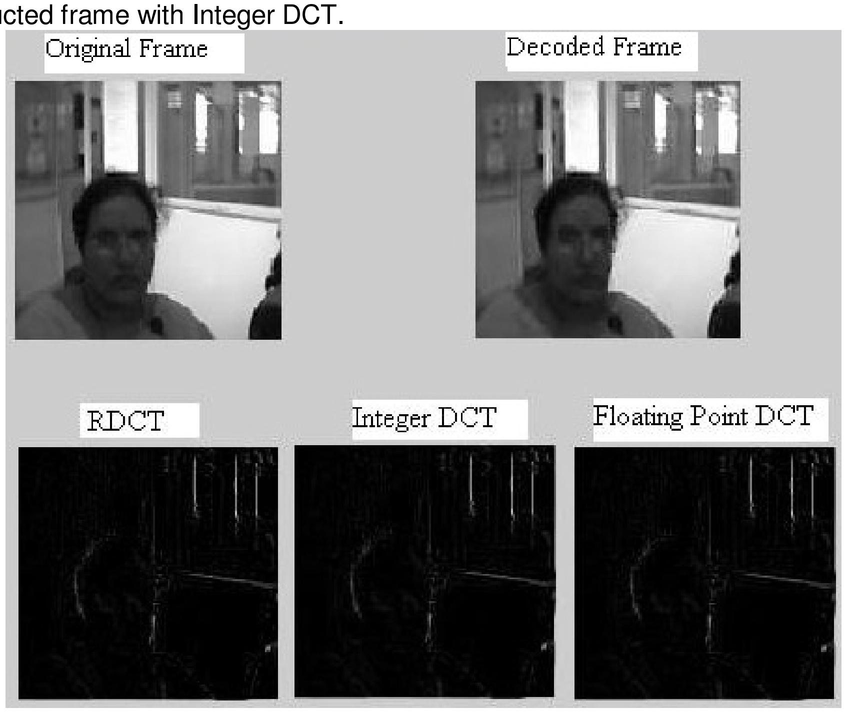

FIGURE 3a Difference between original and decoded frame (real time sequence)

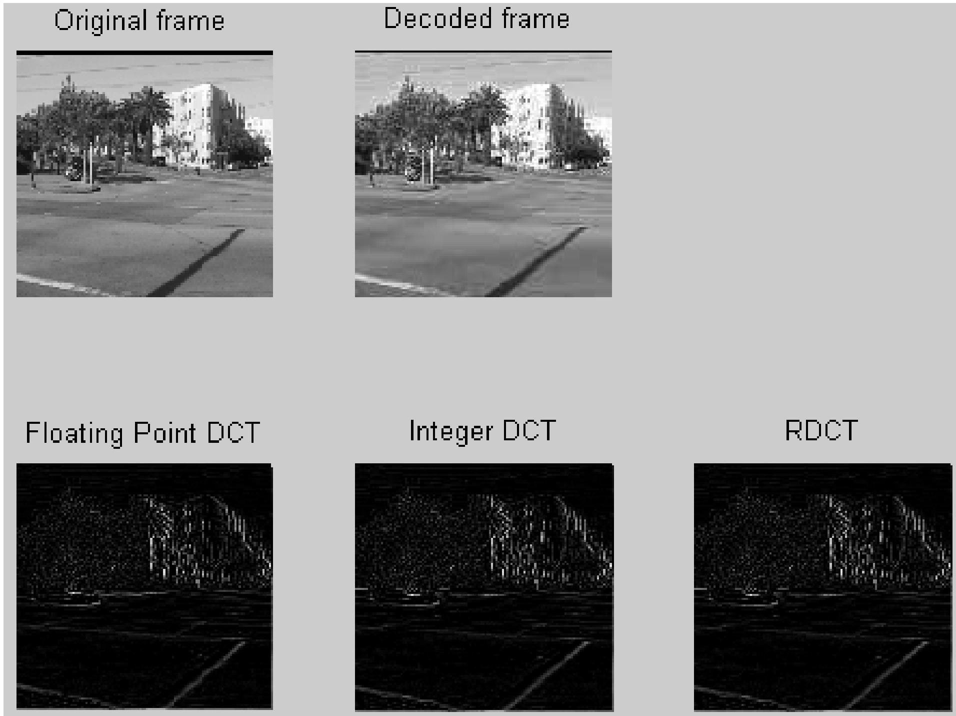

FIGURE 3B Difference between original and decoded frame (San_Fran_Traffic.avi)

Table IV shows the comparison of the computation time for decoding I reference frame and decoding 60 frames, with different algorithms with a GOP of 10 frames. The computation was performed on a Intel Core 2 Duo Processor, @ 1.80 GHz.

TABLE IV DECODING TIME IN SECONDS

Test Sequence	Decoding frame	Floating-point DCT	Integer DCT	Multiplierless RDCT
Real time Data (Frames grabbed at 30 fps)	Reference frame	0.191	0.313	0.125
	10 frames	12.609	14.953	6.562
San_Fran_Traffic	Reference frame	0.296	0.308	0.109
	10 frames	12.322	14.641	6.335
Alex	Reference frame	0.245	0.325	0.125
	10 frames	12.484	14.547	6.593

Table IV shows that the proposed RDCT has reduction in decoding time for 10 frames by 47.9578% when compared with the floating-point DCT whereas it has an improvement of 56.1158% over the commonly used integer DCT for a real time data sequence. However, the reduction in the time is 47.1884% when compared with the floating-point DCT whereas it has an improvement of 54.6779% over integer DCT for a standard data sequence like Alex.avi.

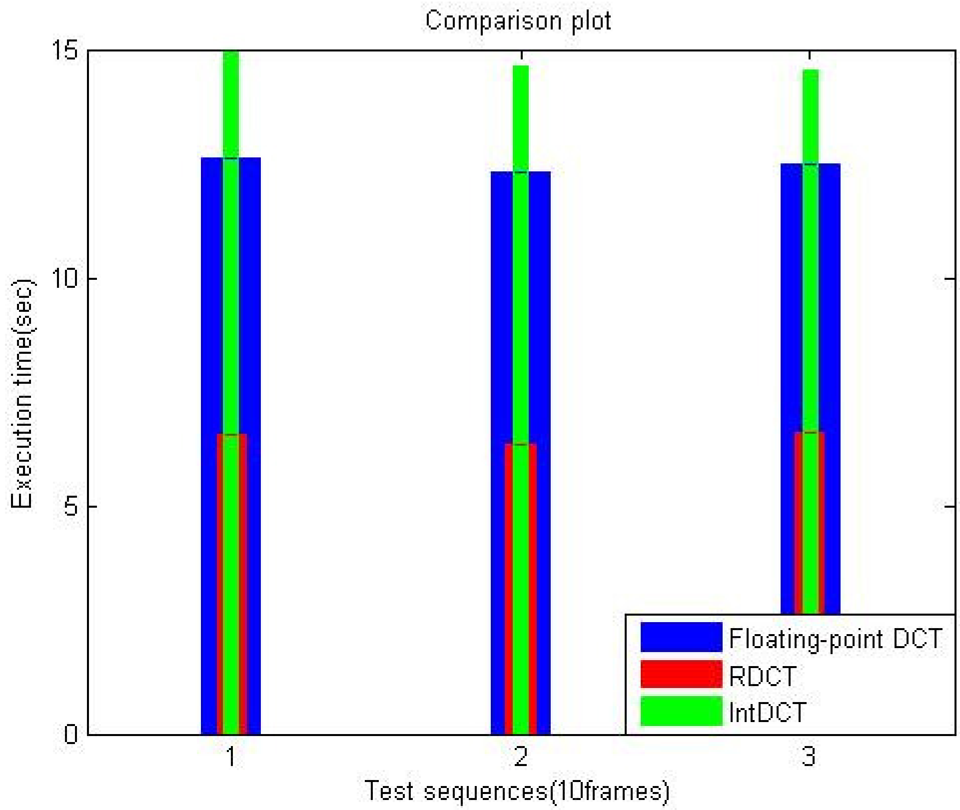

FIGURE 4. Comparison plot for sequences 'Real-time sequence', 'San_Fran_Traffic' & 'Alex'

Fig 4 gives us better comparison in terms of the execution times for decoding 10 frames using different algorithms namely RDCT, floating-point DCT and the IntDCT. The plot clearly shows the RDCT outperforms the floating-point DCT and the IntDCT. This improvement in the decoding time is due to the improvement in the computational complexity of the DCT algorithm.

5. CONCLUSION

The computationally less complex video coding technique is presented in this paper using multiplierless Ramanujan ordered DCT. 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 and adders. This algorithm takes $N/2\log_2 N$ shifts and $(3N/2\log_2 N) - N + 1$ addition operations to evaluate an N-point DCT. The cosine approximation increases the overhead on the number of adders by 13.6% but totally avoids floating point multiplications. The reduction in complexity is reflected in the time required for the decoding of video frames. There is an improvement of 58% from the existing commonly used Integer DCT video codec. The average SSIM and average PSNR values indicate that the quality of decoding using the RDCT is same as that of the Integer DCT. Hence, the proposed algorithm is an efficient multiplierless transform for video coding that offers less computationally complexity but assures the same quality as that of the existing algorithms.

6. REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of ITU-T Recommendation H.261, "Video codecs for audiovisual services at p x 64 kb/s," Mar. 1993.

[2] ITU-T Recommendation H.263, "Video coding for low bitrate communication," Mar. 1996.

[3] ISO/IEC 11 172-2, “Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s: Part 2 Video,” Aug. 1993.

[4] ITU-T Recommendation H.262 I ISODEC 13818-2, “Information technology - generic coding of moving pictures and associated audio information: video,” 1995.

[5] ISO/IEC 13818-2 "Generic Coding of Moving Pictures and Associated Audio Information: Video",

[6] ATSC document A/54 "Guide to the Use of the ATSC Digital Television Standard"

[7] Renxiang Li, Bing Zeng and Ming I.Liou, “A New Three-Step Search Algorithm for Block Motion Estimation”, IEEE Trans. Circuits And Systems For Video Technology, Vol.4, No.4, pp. 438-442, Aug 1994.

[8] Aroh Barjatya, “Block Matching Algorithms For Motion Estimation” , DIP 6620 Spring 2004 Final Project Paper.

[9] 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.

[10] Yonghong Zeng, Lizhi Cheng, Guoan Bi, and Alex C. Kot, ‘‘Integer DCT’s and Fast Algorithms”, IEEE Signal Proc.141-14 (2000).

[11] Geetha.K.S, V.K.Ananthashayana, ‘‘A Novel Recursive Multiplierless Algorithm for 2-D DCT”,Proc. ICSPCN 2009,Aug 2009.

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

[13] Geetha.K.S, M.Uttarakumari, “A Novel Cosine approximation for high-speed evaluation of DCT” International Journal of Image Processing, CSC Journals Volume: 4 Issue: 6 Pg 539 – 548 Jan-Feb 2011.