International Journal of Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.05,Issue.51
December-2016,
Pages:10510-10514

www.ijsetr.com

Implementation of Low Dense Discrete Cosine Transform using Vedic

MEDISETTI S S MANIKANTA[1], D. S. SAMBASIVA RAO[2]

[1]PG Scholar, Dept of ECE, Prasiddha College of Engineering & Technology, Anathavaram, Amalapuram, AP, India.
[2]Associate Professor, Dept of ECE, Prasiddha College of Engineering & Technology, Anathavaram, Amalapuram, AP, India.

Abstract: The main objective of this project is to design a recursive algorithm to obtain an orthogonal approximation of the DCT with half optimized complexity. This project presents a generalized recursive algorithm to obtain an orthogonal approximation of DCT where a pair of DCTs of length N/2 is used to derive approximate DCT of length N at the cost of N additions for input preprocessing. By using symmetries of basis vectors and perform recursive sparse matrix decomposition for deriving the proposed approximation algorithm. The proposed algorithm is highly scalable for hardware as well as software implementation of DCT of larger lengths, and they can be derived using the approximation of existing 8-point DCT to obtain approximate DCT of any power of two length, N>8. Further, this project is enhanced by using Vedic sutras. A technique of binary digits, decimal number multiplication is performed, and it is different from the conventional method of multiplication like Add and Shift. It presents a systematic methodology for high speed and area efficient Vedic Multiplier based on Vedic Mathematics. The multiplier architecture is based on the URDHVA–TIRYAGBYAM sutra of Ancient Indian Vedic Mathematics.

Keywords: Discrete Cosine Transform, Vedic, Approximation Algorithm, Urdhva – Tiryagbyam.

I. INTRODUCTION

The goal of scalable compression methods is to generate a bit string that can be truncated at any desired point, while maintaining the best possible quality (e.g. peak signal-tonoise ratio, PSNR) for the selected bit rate. The availability of such a scalable bit string considerably simplifies the system design by practically eliminating the need for a buffer control method when fitting the data to a certain given bit rate or memory size. In particular, the same single bit string simultaneously serves different channels with different bit rates, without the need to re-encode the original data. Thus, real-time adaptation to varying channel capacities (with application to the Internet or wireless communication channels) is very much simplified. The disadvantage of the well known scalable methods of [1, 2] is their complexity. It turns out, however, that complexity reductions are possible without major losses in performance. For example, the methods of [3, 4, 5] are based on the DCT instead of the wavelet transform, which reduces the complexity of the transform at the cost of a PSNR reduction of 0.6–1 dB [6]. A further complexity reduction for DCT-based scalable compression was achieved in [5], by not making use of trees (similarly, scalable wavelet transform coding without the use of trees was proposed in [7]). An integrated module of contemporary video/image processing applications is constituted by transforming coding: It relies on the basis that pixels in the picture provide a certain level of correlation with the neighboring pixels and adjacent pixels in consecutive video frames show very high correlation in a video transmission system.

Consequently, these correlations can be developed to approximate the value of a pixel from its individual neighbors. A transformation is, therefore, described to plot this spatial, i.e. correlated information into transformed i.e. uncorrelated coefficients. Obviously, the transformation should utilize the fact that the information content of an individual pixel is moderately small i.e. to a large extent visual contribution of a pixel can be estimated using its neighbors. The importance of this paper is to prepare a plan to fix a 8-point Discrete Cosine Transform (DCT) and Inverse DCT with the speed of processing by scaling and approximation of the co-efficient by choosing the proper method of selection of these coefficients. It could be completed by increasing of glided point esteem with contain the outline architecture is designed in Verilog Hardware Description Language code using Modelsim, Altera and XILINX ISE devices. The system is showing and combined using RTL (Register Transfer Level) reflection. In this novel, an 8 × 8 point DCT and IDCT DSP Processor is performed by using Loeffler factorization. The paper gives the data about how to abstain from coasting point by using the DCT/IDCT operations. In this model only 11 duplications are used for implantation. Here the executed configuration is used for the further developments. The pipelined design can likewise be added to DCT and IDCT. The displayed design of processor is combined with the several things which are used as a single processor for the number of applications. The elements of N-point DCT matrix are given by

$$c(i,j) = \epsilon_i \sqrt{\frac{2}{N}} \cos\frac{(2j+1)i\pi}{2N} \quad \textbf{(1)}$$

Where $0 \le i, j \le N-1$, $\epsilon_0 = 0.707$ and $\epsilon_i = 1$ for $I > 0$.the equation is referred to as exact DCT in order to distinguish it from approximated forms of DCT. For $k \in [0,(N/2)-1]$ and $i=2k$, for any even value of N then equation becomes

$$c(2k,j) = \epsilon_{2k} \sqrt{\frac{2}{N}} \cos\frac{(2j+1)2k\pi}{2N} \quad \textbf{(2)}$$

Since $\epsilon_{2k} = \epsilon_k$ the equation can be written as

$$c(2k,j) = \epsilon_k \sqrt{\frac{2}{N}} \cos\frac{(2j+1)k\pi}{N} \quad \textbf{(3)}$$

Hence, the cosine transforms kernel on the right-hand side corresponds to N/2 -point DCT. Therefore, the first N/2 elements of even rows of the DCT matrix of size N × N corresponds to the N/2 -point DCT matrix. Accordingly, the recursive decomposition of C_N can be performed. Using the even/odd symmetries of its row vectors, DCT matrix C_N can be represented by the following matrix product

$$C_N = \frac{1}{\sqrt{2}} M_N^{per} T_N M_N^{add} \quad \textbf{(4)}$$

Where T_N is a block sparse matrix expressed by

$$T_N = \begin{bmatrix} C_{\frac{N}{2}} & 0_{\frac{N}{2}} \\ 0_{\frac{N}{2}} & S_{\frac{N}{2}} \end{bmatrix} \quad \textbf{(5)}$$

Where $0_{N/2}$ is the (N/2× N/2) zero matrixes. Block sub-matrix $S_{N/2}$ consists of odd rows of the first N/2 columns of $\sqrt{2}C_N$. Where M_N^{per} is a permutation matrix expressed by

$$M_N^{per} = \begin{bmatrix} P_{N-1,\frac{N}{2}} & 0_{1,\frac{N}{2}} \\ 0_{1,\frac{N}{2}} & P_{N-1,\frac{N}{2}} \end{bmatrix} \quad \textbf{(6)}$$

Where $0_{1,\frac{N}{2}}$ a row of N/2 is is zeros and $P^{(i)}_{N-1,N/2}$ is a matrix defined by its row vectors as

$$P^{(i)}_{N-1,N/2} = \begin{cases} 0_{1,\frac{N}{2}} & if\ i = 1,3,5,\dots,N-1 \\ I^{(\frac{i}{2})}_{\frac{N}{2}} & if\ i = 0,2,4,\dots..N-2 \end{cases} \quad \textbf{(7)}$$

Where $I_{\frac{N}{2}}(i/2)$ is the (i/2)th row vector of the (N/2 × N/2) identity matrix. Finally, the last matrix M_N^{add} is defined by

$$M_N^{add} = \begin{bmatrix} I_{\frac{N}{2}} & J_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -J_{\frac{N}{2}} \end{bmatrix} \quad \textbf{(8)}$$

Where $J_{N/2}$ is an (N/2 × N/2) matrix having all ones on the anti-diagonal and zeros elsewhere.

To decrease the computational complication of Discrete Cosine Transform, the computational cost of matrices offered is requisite to be measurable. Given that, it does not involve any calculation or logic operation, and requires accompaniments and subtractions, they make a payment very little to the whole arithmetic complexity and cannot be condensed more. So, for declining the computational complexity of N-point DCT, we necessitate to estimate T_N in the equation. Let and denote the approximation matrices of $C_{N/2}$ and $S_{N/2}$, respectively. To find these approximated sub matrices we take the smallest size of the DCT matrix to terminate the approximation procedure to 8, because four-point DCT and two-point DCT can be implemented with adders simply. Consequently, a good approximation of C_N, where N is an integral power of two, for $N \ge 8$, leads to proper approximations of C_8 and S_8. For an approximation of C_8 we can choose the 8-point DCT. Since that presents the best exchange stuck between the number of necessary arithmetic operators and quality of the reconstructed image.

II. C_8 IMPLEMENTATION

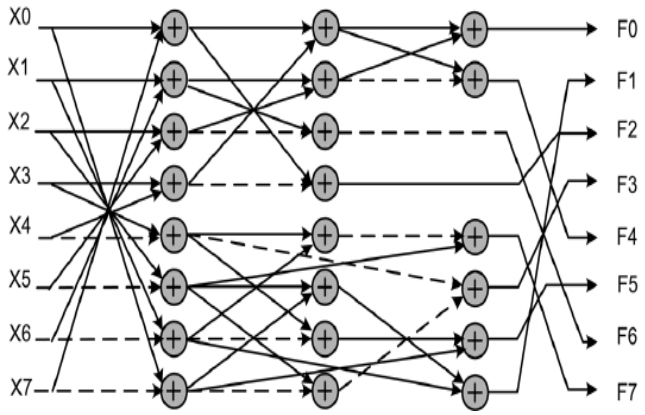

Fig.1.Signal Flow Graph (sfg) of (c8).

The basic computational block of algorithms for the existing DCT approximation, The block diagram of the computation of DCT based on C8 is shown in above fig.1, for a given input sequence {X (n)}, $n \in [0, N-1]$. The approximate DCCT coefficients are obtained by F= . X^T. Can be approximated by two units for the computation of are used along with an input adder unit and output permutation unit. And computation of 32-point DCT could be obtained by combining a pair of 16-point DCTs with an input adder block and output permutation block. To assess the computational complexity of existing N-point approximate DCT, we need to determine the computational cost of matrices, the approximate 8-point DCT involves 22 additions. Since permutation matrix has no computational cost and addition matrix requires additions for N additions for N-point DCT, the overall arithmetic complexity of 16-point, 32-point, and 64-point DCT approximations are 60, 152, and 368 additions, respectively. More generally, the arithmetic complexity of N-point DCT is equal to N ($\log_2$ N-(1/4)) additions. Moreover, since the structures for the calculation of Discrete Cosine Transform of dissimilar lengths are normal and scalable, the calculation time for N-point DCT coefficients can be determined to be $\log_2(N)T_A$,where T_A is the extra-time. The method requires the least number of augmentations, and does

not need any shift functions. Make a note of that shift process does not involve any combinational components, and need simply as rewiring during hardware execution. Other than it has oblique giving to the hardware complication because shift add operations direct to increase in bit-width which leads to higher hardware density of arithmetic units which go after the shift-add operation. Also, I note that all measured estimate methods involve extensively less computational complexity over that of the exact DCT algorithms**.**

III. EXISTING RECONFIGURABLE ARCHITECTURE

Discrete Cosine Transform (DCT) of dissimilar lengths such as 32, 16 are needed to be used in video coding applications. So, a known Discrete Cosine Transform structural design have to be potentially reused for the DCT of altered lengths instead of using different designs for different lengths. I suggest here such reconfigurable DCT designs which could be reused for the calculation of DCT of different lengths. The reconfigurable structural design for the functioning of approximated 16-point DCT is shown in below Fig.2.

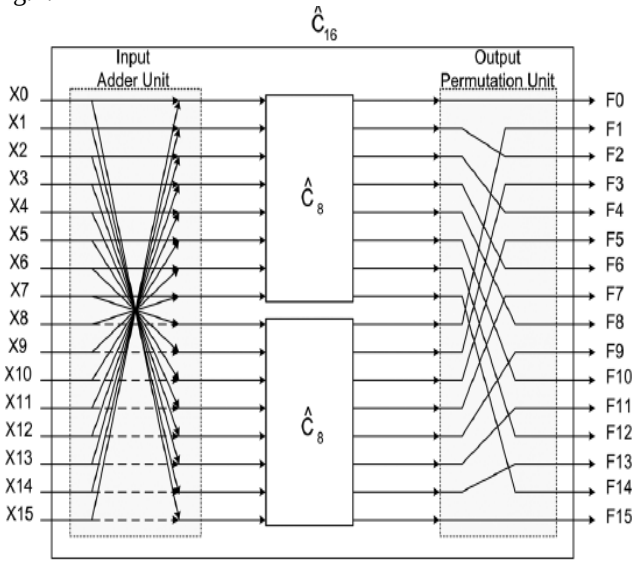

Fig.2.Block diagram of approximation of DCT for n=16.

It consists of three computing units, that is 2 eight-point approximated Discrete Cosine Transform units and a sixteen-point input adder unit that generates a(i) and b(i),I ϵ[1:7] . The input to the initial 8-point DCT approximation unit is fed through 8 MUXes that select either a(0), a(1), a(2) ,a(3) ,a(4) ,a(5) a(6), a(7) or X(0),X(1),X(2),X(3),X(4), X(5),X(6),X(7) depending on whether it is used for 16-point DCT calculation or 8-point DCT calculation. Similarly, the input to the second 8-point DCT unit is fed through 8 MUXes that select either b(0),b(1),b(2),b(3),b(4),b(5),b(6),b(7) or X(8),X(9),X(10), X(11),X(12),X(13), X(14) ,X(15), based on whether it is used for sixteen-point DCT calculation or eight-point DCT calculation. The unit uses 14 MUXes to select and re-order the output depending on the size of the selected DCT.sel16 is used as control input of the MUXes to select inputs and to perform permutation. Specifically, sel16=1 enables the computation of 16-point DCT and enables the computation of a pair of 8-point DCTs in parallel. Consequently, the architecture allows the calculation of a 16-point DCT or two 8-point DCTs in parallel. A reconfigurable design for the computation of 32-, 16-, and 8-point DCTs is presented.

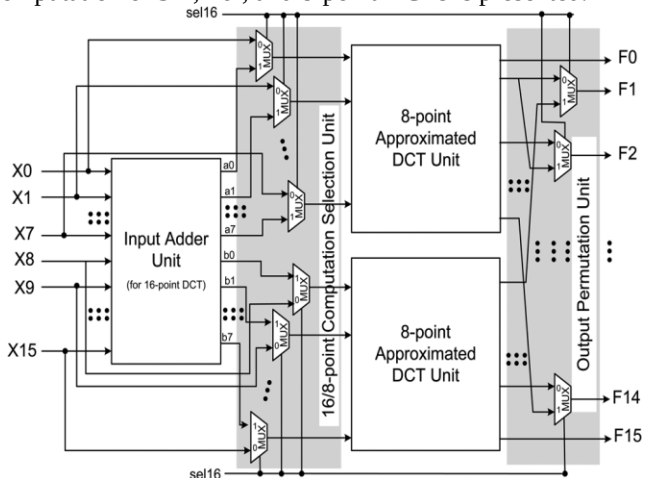

Fig.3. Reconfigurable Architecture for DCT of lengths n=8 & 16.

It performs the computation of a 32-point Discrete Cosine Transform or 2 sixteen-point Discrete Cosine Transforms in parallel or 4 eight-point Discrete Cosine Transforms in parallel. The structural design is poised of 32-point input adder unit, 2 sixteen-point input adder units, and 4 eight-point DCT units. The reconfigurability is accomplished by 3 control blocks unruffled of 64 2:1 MUXes along with 30 3:1 MUXes as shown in Fig.3. The primary control block decides whether the Discrete Cosine Transform size is of 32 or lower. If, the selection of input data has ended for the 32-point Discrete Cosine Transform, or else, for the Discrete Cosine Transforms of lower lengths. The second control block decides whether the Discrete Cosine Transform size is higher than 8.

IV. DCT USING VEDIC MULTIPLIER

The hardware, structural design of 2×2, 4x4 and 8x8 bit Vedic multi-player components are shown in the lower sections. At this point, "URDHVA-TIRYAGBHYAM" (perpendicularly and diagonally) sutra is used to suggest such design for the multiplication of two binary information. The attractiveness of Vedic multiplier is that here partial product generation and additions are done simultaneously. Therefore, it is well modified to equivalent processing. The attribute make it more beautiful for binary multiplications. This in turn decline, delay, which is the main inspiration following this work. A Vedic Multiplier for 2x2 bit Module, The process is explained below for two, 2 bit numbers X and Y where X = a1a0 and Y = b1b0. Initially, the slightest significant bits are multiplied which gives the least significant bit of the ultimate product (vertical). Then, the Least Significant Bit of the multiplicand is multiplied with the next superior bit of the multiplier and added to, the product of Least Significant Bit of multiplier and a next higher bit of the multiplicand (crosswise). The sun produces second bit of the final product and the carry is further with the partial product

get hold of by multiplying the most significant bits to give the sum and carry. The sum is the third equivalent bit and carry becomes the fourth bit of the final product as shown in Fig.4.

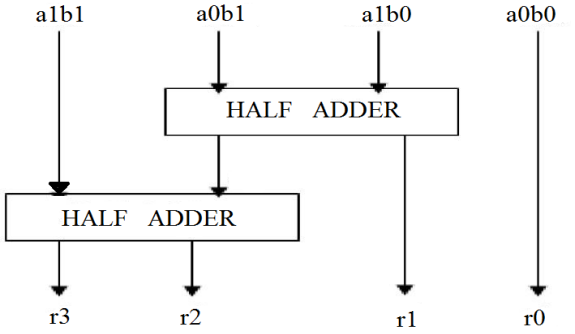

Fig.4. Block diagram 2×2 Vedic multiplier.

The 2X2 Vedic multiplier unit is realized with 4 input AND gates and 2 half-adders which is shown in its block. It is originate that the hardware structural design of 2x2 bit Vedic multiplier unit is same as the hardware architecture of 2x2 the enhancemental Array Multiplier. Therefore, it is over and done with that multiplication of two bit double numbers by Vedic sutra does not ended important to effect in the enhancement of the multiplier's effectiveness. Extremely state that the whole delay is only two-half adder delays, later than final bit products are produced, which is alike to Array multiplier. So we switch more than the implementation of 4x4 bit Vedic multiplier which uses the 2x2 bit multiplier as a fundamental building block. The identification method can be extensive for input bits 4 & 8. But for larger number of bits in input, little alteration is necessary. The 4x4 bit Vedic multiplier unit is designed by four 2x2 bit Vedic multiplier units. Let's examine 4x4 calculations, say X= A3A2A A0 and Y= B3 B2 B1 B0. The output line for the multiplication unit is - S7 S6 S5 S4 S3 S2 S1 S0. Let's split X and Y into two parts, say A3A2 & A1A0 for X and B3B2 & B1B0 for Y. With the basic unit of Vedic multiplication, taking 2 bits at a time and using 2 bit multiplier units, we can have the subsequent design for calculation as displayed in below fig.5. Model representation for 4x4 bit Vedic Multiplication has each block as shown top is 2x2 bit Vedic multiplier unit.

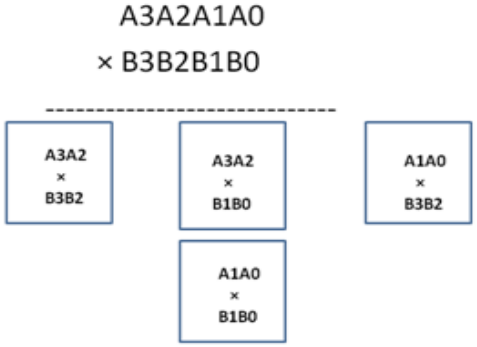

Fig.5. A sample representation of 4 bit Vedic multiplication.

Primary 2x2 bit multiplier unit inputs are A1A0 and B1B0. The final block is 2x2 bit multiplier unit with inputs A1A2 and B3B2.The center one display 2 2x2 bit multiplier units with inputs A3A2 & B1B0 and A1A0 & B3 B2. Consequently the ending result of multiplication, which is eight bits, S7, S6 S5 S4 S3 S2 S1 S0. To obtain the last product (S7 S6 S5 S4 S3 S2 S1 S0), four 2x2 bit multiplier units and three 4-bit Ripple-Carry Adder units are required. The planned Vedic multiplier unit can be second-hand to reduce delay. In the early hours novel speak about Vedic multiplier units are based on array multiplier designs. On the other hand, I projected a new architecture, which is proficient in terms of speediness useful to reduce delay. Grippingly, an 8x8 Vedic multiplier segment is developed easily with four 4x4 multiplier units as shown in Fig.6.

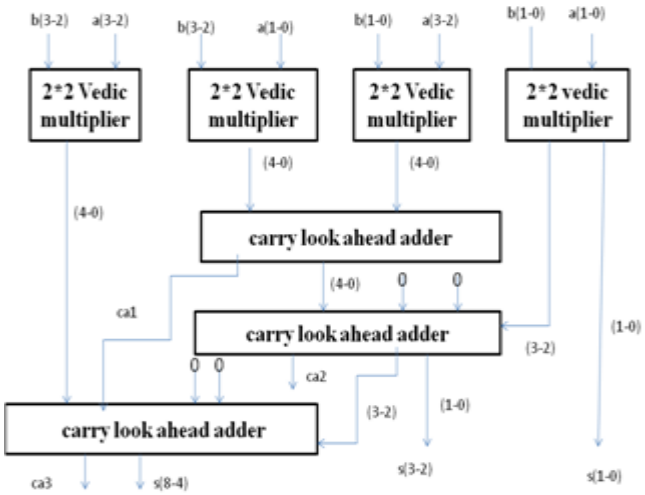

Fig.6. Block diagram for 4 bit Vedic multiplication.

V. RESULT

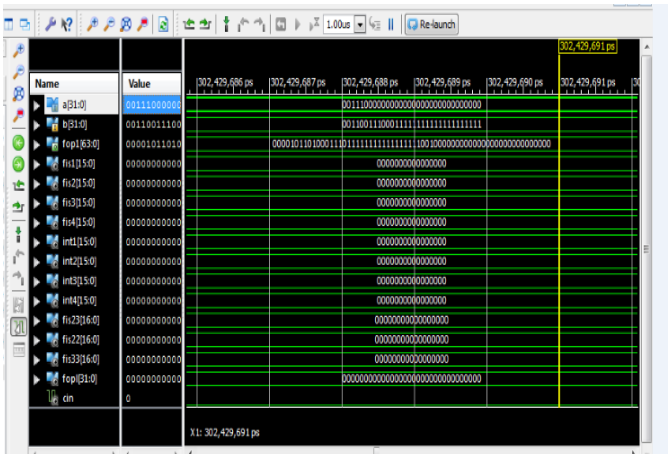

Fig.7. Simulation Waveform.

VI. CONCLUSION

In this paper, presented a discrete cosine transform by employing Vedic multiplier architecture of URDHVA - TIRYAGBYAM sutras. Whereas the existing design is modified by multiplier architecture using carry look-ahead adder for a reduced amount of propagation delay, with a reduction of power consumption and area efficient by reducing the number of components. The key idea was to provide an increase of processing speed and save the time in high throughput applications.

VII. REFERENCES

[1]Maher Jridi, Ayman Alfalou,and Pramod Kumar Maher IEEE“ a generalized algorithm and reconfigurable architecture for Efficient and Scalable Orthogonable Approximation of DCT, IEEE Transactions On circuits and Systems, Vol. 62, no 2 FEBRUARY 2015.

[2]A. M. Shams, A. Chidanandan,W. Pan, and M. A. Bayoumi, “NEDA:A low-power high-performance DCT architecture,” IEEE Trans.Signal Process., vol. 54, no. 3, pp. 955–964, 2006.

[3]C.Loeffler, A.Lightenberg, and G.S. Moschytz, “Practical fast 1-DDCT algorithm with 11 multiplications,” in Proc. Int. Conf. Acoust.,Speech, Signal Process. (ICASSP), May 1989, pp. 988–991.

[4]M. Jridi, P. K. Meher, and A. Alfalou, “Zero-quantised discrete cosine transform coefficients prediction technique for intra-frame video encoding,”IET Image Process., vol. 7, no. 2, pp. 165–173, Mar. 2013.

[5]S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, “Binary discrete cosine and Hartley transforms,” IEEE Trans. Circuits Syst. I, Reg. Papers,vol. 60, no. 4, pp. 989–1002, Apr. 2013.

[6]F. M. Bayer and R. J. Cintra, “DCT-like transform for image compression requires 14 additions only,” Electron. Lett., vol. 48, no. 15, pp.919–921, Jul. 2012.

[7]R. J. Cintra and F. M. Bayer, “A DCT approximation for image compression,”IEEE Signal Process. Lett., vol. 18, no. 10, pp. 579–582,Oct. 2011.

[8] S. Bouguezel, M. Ahmad, and M. N. S. Swamy, “Low-complexity 8×8 transform for image compression,” Electron. Lett., vol. 44, no. 21,pp. 1249–1250, Oct. 2008.

[9] T. I. Haweel, “A new square wave transform based on the DCT,” Signal Process., vol.81, no. 11, pp. 2309–2319, Nov. 2001.

[10] V. Britanak, P.Y.Yip, and K. R. Rao, Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations. London, U.K.: Academic, 2007.

[11] G. J. Sullivan, J.-R. Ohm,W.-J.Han, and T.Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no.12, pp. 1649–1668, Dec. 2012.

[12] F. Bossen, B. Bross, K. Suhring, and D. Flynn, “HEVC complexity and implementation analysis,” IEEE Trans. Circuits Syst. Video Technol.,vol. 22, no. 12, pp. 1685–1696, 2012.

[13] X. Li, A. Dick, C. Shen, A. van den Hengel, and H. Wang, “Incremental learning of 3D-DCT compact representations for robust visual tracking,” IEEE Trans. Pattern Anal.Mach. Intell., vol. 35, no. 4, pp.863–881, Apr. 2013.

[14] A. Alfalou, C. Brosseau, N. Abdallah, andM. Jridi, “Assessing the performance of a method of simultaneous compression and encryption of multiple images and its resistance against various attacks,” Opt. Express, vol. 21, no. 7, pp. 8025–8043, 2013.

[15] R. J. Cintra, “An integer approximation method for discrete sinusoidal transforms,” Circuits, Syst., Signal Process., vol. 30, no. 6, pp.1481–1501, 2011.

[16] F. M. Bayer, R. J. Cintra, A. Edirisuriya, and A. Madanayake, “A digital hardware fast algorithm and FPGA-based prototype for a novel 16-point approximate DCT for image compression applications,”Meas. Sci. Technol., vol. 23, no. 11, pp. 1–10, 2012.

[17]R. J. Cintra, F. M. Bayer, and C. J. Tablada, “Low-complexity 8-point DCT approximations based on integer functions,” Signal Process., vol.99, pp. 201–214, 2014.

[18] U. S. Potluri, A.Madanayake, R. J. Cintra, F.M. Bayer, S.Kulasekera, and A. Edirisuriya,“Improved 8-point approximate DCT for image and video compression requiring only 14 additions,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 6, pp. 1727–1740, Jun.2014.