

Orthogonal Approximation of DCT in Video Compressing Using Generalized Algorithm

J.Sindhukavi¹, J. Ancy Finea², A. Josephine Sugan Priya³, K. Solaiyammal⁴

^{1,2,3,4}ECE Department, Idhaya Engineering College for Women, Chinnasalem, Villupuram, Tamil Nadu, India

ABSTRACT

Approximation of Discrete cosine transform (DCT) is useful for reducing its computational complexity without significant impact on its coding performance. Most of the existing algorithms for approximation of the DCT target only the DCT of small transform lengths, and some of them are non-orthogonal. We perform recursive sparse matrix decomposition and make use of the symmetries of DCT basis vectors for deriving the proposed approximation algorithm. Proposed algorithm is highly scalable for hardware as well as software implementation of DCT of higher length. We demonstrate that the proposed approximation of DCT provides comparable or better image and video compression performance than the existing approximation methods. It is shown that proposed algorithm involves lower arithmetic complexity compared with the other existing approximation algorithms. We have presented a fully scalable reconfigurable parallel architecture for the computation of approximateDCT based on the proposed algorithm. One uniquely interesting feature of the proposed design is that it could be configured for the computation of a 32-point DCT or for parallel computation of two 16-pointDCTs or four 8-point DCTs with a marginal control overhead. The proposed architecture is found to offer many advantages in terms of hardware complexity, regularity and modularity. Experimental results obtained from FPGA implementation show the advantage of the proposed method.

Keywords:Algorithm-Architecture Code Sign, DCT Approximation, Discrete Cosine Transform (DCT), high Efficiency Video Coding (HEVC)

I. INTRODUCTION

The discrete cosine transform (DCT) is popularly used inimage and video compression. Since the DCT is computationally intensive, several algorithms have been proposed in the literature to compute itefficiently. Recently, significant work has beendone to derive approximate of 8-point DCT for reducing the computational complexity [4]–[9].

The main objective of the approximation algorithms is to get rid of multiplications which consume most of the power and computation- time, and to obtain meaningful estimation of DCT as well. Haweelhas proposed the signed DCT (SDCT) for 8-8 Blocks where the basis vector elements are replaced by their sign, i.e,Bouguezel-Ahmad-Swamy (BAS) have proposed a series of methods. They have provided a good estimation of the DCT by replacing the basis vector elements by 0, 1/2, 1. In the same vein, Bayer and Cintra have proposed two transforms derived from 0 and 1 as elements of transform kernel, and have shown that their methods perform better than the method in, particularly for low- and high-compression ratio scenarios. The need of approximation is more important for higher-size DCT since the computational complexity of the DCT grows nonlinearly.

On the otherhand, modern video coding standards such as high efficiency video coding (HEVC) [10] uses DCT of larger block sizes (up to 32) in order to achieve higher compression ratio. But, the extension of the design strategy used in H264 AVC for larger transform sizes, such as 16-point and 32-point is not possible.

Besides, several image processing applications such as tracking and simultaneous compression and encryption require higher DCT sizes. In this context, Cintra has introduced a new class of integer transforms applicable toseveral block-lengths.A scheme of approximation of DCT should have the following features:

- i. It should have low computational complexity.
- ii. It should have low error energy in order to provide compression performance close to the exact DCT and preferably should be orthogonal.
- iii. It should work for higher lengths of DCT to support modern video coding standards and other applications like tracking, surveillance and simultaneous compression and encryption.

The proposed approximate form of DCT of different lengths are orthogonal and result in lower error-energy compared to the existing algorithms for DCT approximation. The decomposition process allows generalization of the proposed transform for higher-size DCTs. Interestingly, proposed algorithm is easily hardware as well scalable for as software implementation of DCT of higher lengths and it can make use of the best of the existing approximations of 8-point DCT. Based on the proposed algorithm, we have proposed a fully scalable, reconfigurable and parallel architecture for approximate DCT computation.

II. METHODS AND MATERIAL

1. Literature Survey

SatyajayantMisra, Martin Reisslein and GuoliangXue have proposed a wireless sensor network with multimedia capabilities typically consists of data sensor nodes, which sense, for instance, sound or motion and video sensor nodes, which capture video of events of interest. In this survey, we focus on the video encoding at the video sensors and the real-time transport of the encoded video to a base station. Real-time video streams have stringent requirements for end-to-end delay and loss during network transport. In this survey, we categorize the requirements of multimedia traffic at each layer of the network protocol stack and further classify the mechanisms that have been proposed for multimedia streaming in wireless sensor networks at each layer of the stack. Specifically, we consider the mechanismsoperating at the application, transport, network, and MAC layers.

And alsoKrisdaLengwehasatit have proposed the discrete cosine transform (DCT) is one of the major

components in most of image and video compression systems. The variable complexity algorithm framework has been applied successfully to achieve complexity savings in the computation of the inverse DCT in decoders. These gains can be achieved due to the highly predictable sparseness of the quantized DCT coefficients in natural image/video data. With the increasing demand for instant video messaging and two-wav video transmission over mobile communication systems running ongeneral-purpose embedded processors, the encoding complexity needs to be optimized. In this paper, we focus on complexity reduction techniques for the forward DCT, which is one of the more computationally intensive tasks in the encoder. Unlike the inverse DCT, the forward DCT does not operate on sparse input data, but rather generates sparse output data. Thus, complexity reduction must be obtained using different methods from those used for the inverse DCT. In the literature, two major approaches have been applied to speed up the forward DCT computation, namely, frequency selection, in which only a subset of DCT coefficients is computed and accuracy selection, in which all the DCT coefficients are computed with reduced accuracy. approaches can achieve significant These two computation savings with minor output quality degradation, as long asthe coding parameters are such that the quantization error is larger than the error due to the approximate DCT computation.

The application of several families of fast multiplier less approximations of the discrete cosine transform (DCT) with the lifting scheme called the bin DCT. These bin DCT families are derived from Chen's and Loeffler's plane rotation-based factorizations of the DCT matrix, respectively and the design approach can also be applied to a DCT of arbitrary size. Two design approaches are presented. In the first method, an optimization program is defined, and the multiplier less transform is obtained by approximating its solution with dyadic values.

In the second method, a general lifting-based scaled DCT structure is obtained, and the analytical values of all lifting parameters are derived, enabling dynamic approximations with different accuracies. Therefore, the bin DCT can be tuned to cover the gap between the Walsh–Hadamard transform and the DCT. The corresponding two-dimensional (2-D) bin DCT allows a 16-bit implementation, enables lossless compression,

and maintains satisfactory compatibility with the floating-point DCT. The performance of the bin DCT in JPEG, H.263+ and lossless compression is also demonstrated.

2. Existing System

In the mathematical description of the selected 8-point DCT approximations. All discussed methods here consist of a transformation matrix that can be put in the following format: [Diagonal matrix] * [Low complexity matrix] The diagonal matrix usually contains irrational numbers in the form 1/, where m is a small positive integer. In principle, the irrational numbers required in the diagonal matrix would require an increased computational complexity. The existing algorithms are, \Box BAS-2008 approximate DCT (T1)
BAS-2011 approximate DCT (Ta) CB-2011 approximate DCT (T2) Modified CB-2011 approximate DCT (T3) Approximate DCT (T4). In the context of image compression, the diagonal matrix can simply be absorbed into the quantization step of JPEG-like compression procedures . Therefore, in this case, the complexity of the approximation is bounded by the complexity of the low-complexity matrix. Since the entries of the low complexity matrix comprise only powers of two in $\{0,\pm 1/2,\pm 1,\pm 2\}$, null multiplicative complexity is achieved. Since the existing system is low area efficient, high power consuming and induces high delay we go for our proposed work.

III. RESULTS AND DISCUSSION

1. Proposed System

The proposed system here such reconfigurable DCT structures which could be reused for the computation of The reconfigurable DCT of different lengths. architecture for the implementation of approximated 16-point DCT is shown in Fig. It consists of three computing units, namely two 8-point approximated DCT units and a 16-point input adder unit that generates. The input to the first 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])(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 (Fig.) 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[1 0],x[11],x[12],x[13],x[14],x[15]), depending on whether it is used for 16-point DCT calculation or 8-

point DCT calculation. On the other hand, the output permutation 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 according to the size of the DCT to be computed. Specifically, SEL16=1 enables the computation of 16-point DCT and SEL16=0 enables the computation of a pair of 8-point DCTs in parallel. Consequently, the architecture of Fig. 3 allows the calculation of a 16-point DCT or two 8point DCTs in parallel.

3. MODULE'S:

- ➢ 8-POINT DCT
- ➢ 16 BIT ADDER UNIT
- ➢ 16 POINT DCT
- ➢ 32 POINT DCT

MODULE DESCRIPTION

8-POINT DCT

We closely look at, we note that operates C_8 on sums of pixel pairs while S_8 operates on differences of the same pixel pairs. Therefore, if we replace S_8 by C_8 , we shall have two main advantages. Firstly, we shall have good compression performance due to the efficiency of and secondly the implementation will be much simpler, scalable and reconfigurable. For approximation of S_8 we have investigated two other low-complexity alternatives and in the following we discuss here three possible options of approximation.

- i. The first one is to approximate by null matrix, which implies all even-indexed DCT coefficients are assumed to be zero. The transform obtained by this approximation is far from the exact values of even-indexed DCT coefficients and the odd coefficients do not contain any information.
- ii. The second solution is obtained by approximating S_8 by an 8x8 matrix where each row contains one 1 and all other elements are zeros. Here, elements equal to 1 correspond to the maximum of elements of the exact DCT in each row. The approximate transform in this case is closer to the exact DCT than the solution obtained by null matrix.

iii. The third solution consists of approximation of S_8 by C_8 . Since as C_8 well as S_8 are sub-matrices

of C₁₆ and operate on matrices generated by sum and differences of pixel pairs at distance of 8, approximation of S_8 by C_8 has attractive computational properties: regularity of the signalgraph, orthogonality flow since C_8 is orthogonalizable and good compression efficiency, other than scalability and scope for reconfigurable implementation.



Input Adder Unit:

To assess the computational complexity of proposed point approximate DCT, we need to determine the computational cost of matrices quoted in (9). As shown in Fig. 1 the approximate8-point DCT involves 22 additions. Since has no computational cost and requireadditions for -point DCT, the overall arithmetic complexity of 16-point, 32-point, additions, respectively. More generally, the arithmetic complexity of -point DCT is equal to additions. Moreover, since the structures for the computation of DCT of different lengths are regular and scalable, the computational time for DCT coefficients can be found to be where is the addition-time. The number of arithmetic operations involved in proposed DCT approximation of different lengths and those of the existing competing approximations are shown in Table I. It can be found that the proposed method requires the lowest number of additions and does not require any shift operations. Note that shift operation does not involve any combinational components and requires only rewiring during hardware implementation. But it has indirect contribution to the hardware complexity since shift-add operations lead to increase in bit-width which leads to higher hardware complexity of arithmetic units which follow the shift-add operation. Also, we note that all involve considered approximation methods significantly less computational complexity over that of the exact DCT algorithms. According to the Loeffler algorithm, the exact DCT computation requires 29, 81, 209, and 513 additions along with 11, 31, 79, and191 multiplications, respectively for 8, 16, 32, and 64-point DCTs.



16-POINT DCT

Pipelined and non-pipelined designs of different methods are developed, synthesized and validated using an integrated logic analyzer. The validation is carried out by using the digilent EB of Spartan6-LX45. We have used 8-bit inputs, and we have allowed the increase of output size (without any truncations). For the 8-point transform of Fig. 1, we have 11-bit and 10bit outputs. The pipelined design are obtained by insertion of registers in the input and output stages along with registers after each adder stage, while the no pipeline registers are used within the non-pipelined designs. The synthesis results obtained from XST synthesizer are presented in Table II. It shows that designs significantly pipelined provide higher maximum operating frequency (MOF). It also shows that the proposed design involves nearly 7%, 6%, and 5% less area compared to the BDCT design for equal to 16, 32, and 64, respectively. Note that both pipelined and non-pipelined designs involve the same number of LUTs since pipeline registers do not require additional LUTs. For 8-point DCT, we have used the approximation proposed in which forms the basic computing block of the proposed method. Also, we underline that all designs have the same critical path; and accordingly have the same MOFs. Most importantly, the proposed designs are reusable for different transform lengths.

32-POINT DCT

As specified in the recently adopted HEVC, DCT of different lengths such as N=8,16,32 are required to be used in video coding applications. Therefore, a given DCT architecture should be potentially reused for the DCT of different lengths instead of using separate structures for different lengths. We propose here such reconfigurable DCT structures which could be reused for the computation of DCT of different lengths. The reconfigurable architecture for the implementation of approximated 16-point DCT is shown in Fig. It consists of three computing units, namely two 8-point approximated DCT units and a 16-point input adder unit that generates.

The input to the first 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 (Fig.) 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[1 0],x[11],x[12],x[13],x[14],x[15]), depending on whether it is used for 16-point DCT calculation or 8-point DCT calculation. On the other hand, these output permutation 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 according to the size of the DCT to be computed. Specifically, SEL16=1 enables the computation of 16-point DCT and SEL16=0 enables the computation of a pair of 8-point DCTs in parallel. Consequently, the architecture of Fig.

3 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 in Fig. It performs the calculation of a 32-point DCT or two 16-point DCTs in parallel or four 8-point DCTs in parallel. The architecture is composed of 32-point input adder unit, two 16-point input adder units, and four 8-point DCT units. The reconfigurability is achieved by three control blocks composed of 64 2:1 MUXes along with 30 3:1 MUXes. The first control block decides whether the DCT size is of 32 or lower. If SEL32=1, the selection of input data is done for the 32-point DCT, otherwise, for the DCTs of lower lengths. The second control block decides whether the DCT size is higher than 8. If SEL16=1 the length of the DCT to be computed is higher than 8 (DCT length of 16 or 32), otherwise, the length is 8. The third control block is used for the output permutation unit which re-orders the output depending on the size of the selected DCT.SEL32 and SEL16 are used as control signals to the 3:1 MUXes. Specifically, for {SEL32,SEL16} equal to {00}, {01} or {11} the 32 outputs correspond to four 8-point parallel DCTs, two parallel 16-point DCTs, or 32-point DCT, respectively. Note that the throughput is of 32 DCT coefficients per cycle irrespective of the desired transform size.

2. Graphical Representation and Result

Thus proposed system uses a recursive algorithm toobtain orthogonal approximation of DCT where approximate DCT of length could be derived from a pair.

Thus proposed system uses a recursive algorithm toobtain orthogonal approximation of DCT where approximate DCT of length could be derived from a pair of DCTs of length at the cost of additions for input preprocessing. The proposed approximated DCT has several advantages, such as of regularity, structural simplicity, lower-computational complexity, and Comparison with recently proposed scalability. competing methods shows the effectiveness of the proposed approximation in terms of error energy, hardware sources consumption, and compressed image quality.



We have also proposed a fully scalable reconfigurable architecture for approximate DCT computation where the computation of 32-point DCT could be configured for parallel computation of two 16-point DCTs or four 8-point DCTs.



IV. CONCLUSION

In this paper, we have proposed a recursive algorithm to obtain orthogonal approximation of DCT where approximate DCT of length N could be derived from a pair of DCTs of length (N/2) at the cost of N additions for input preprocessing. The proposed approximated DCT has several advantages, such as of regularity, structural simplicity, lower-computational complexity, and scalability. We have also proposed a fully scalable reconfigurable architecture for approximate DCT computation where the computation of 32-point DCT could be configured for parallel computation of two 16point DCTs or four 8-point DCTs.

V. REFERENCES

- A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, "NEDA: A low-power highperformance DCT architecture," IEEE Trans.Signal Process., vol. 54, no. 3, pp. 955– 964, 2006.
- [2]. C. Loeffler, A. Lightenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithm with 11 multiplications," in Proc. Int. Conf. Acoust., Speech, SignalProcess. (ICASSP), May 1989, pp. 988–991.
- [3]. M. Jridi, P. K. Meher, and A. Alfalou, "Zeroquantised discrete cosine transform coefficients prediction technique for intra-frame video encoding," IET Image Process., vol. 7, no. 2, pp. 165–173, Mar. 2013.
- [4]. 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.
- [5]. 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.
- [6]. 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.
- 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.