2016 IEEE Second International Conference on Multimedia Big Data

A Fast and Lossless IDCT Design for AVS2 Codec

Kaili Yao, Ronggang Wang, Zhenyu Wang, Wenmin Wang, Wen Gao

School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University
National Engineering Laboratory of Video Technology, Peking University
Shenzhen, China
Email: kellyyao@sz.pku.edu.cn, rgwang@pkusz.edu.cn

***Abstract*—The emerging ultra-HD video content and the latest generation video coding standards such as HEVC and AVS2 involve significant computational complexity increase. Aiming to reduce the complexity of IDCT process, a fast IDCT design for AVS2 is presented in this paper, which bases on skipping the calculations of zero coefficients. After statistical analysis, we design several patterns for transform blocks with different sizes to detect the distribution of non-zero coefficients. If a transform block conforms to one of our designed patterns, the corresponding simplified IDCT function will be executed. Experimental results showed that our strategy could reduce the computation time by 19.3% on average under various test conditions. Moreover, our method will not result in any coding performance loss.**

***Index Terms*—Inverse Transform, Inverse Discrete Cosine Transform, Video Compression, AVS2.**

I. Introduction

In order to efficiently compress the emerging ultra-HD video content, the second generation of Audio Video coding Standard (AVS2) was developed by China Audio Video Coding Standard Working Group [1]. It was also issued as IEEE 1857.4 [2]. AVS2 doubles the coding efficiency of previous standards such as AVS1 and H.264/AVC. The coding performance of AVS2 is similar to HEVC [3] for video broadcasting contents, while AVS2 can provide more efficient compression for certain video applications such as surveillance and low-delay communication (e.g., videoconferencing). Moreover, AVS2 is making video coding smarter by adopting intelligent coding tools that not only improve coding efficiency but also help with computer vision tasks such as object detection and tracking. The same to other standards, AVS2 employed the block-based hybrid coding framework. A series of advanced coding tools were adopted in AVS2. For example, the coding units are not limited to quad-tree structure, and both symmetric and asymmetric partitions can be used in prediction units. The size of transform block is varied from 4×4 to 64×64. However, these coding tools involve significant computational complexity increase as well.

In image and video coding process, transform is a key step for providing energy compaction by converting the signals from spatial domain to frequency domain. Since DCT has a distinct advantage of both computability and compaction performance over other transforms, most video compression standards, such as H.26x and MPEG-x, employ the block transform based on DCT (Discrete Cosine Transform) / IDCT (Inverse Discrete Cosine Transform). Owing to matrix multiplication, DCT and IDCT involve a large amount of calculations, so they are still computation sensitive modules in the video codec. In this paper, we focus on the acceleration for IDCT process.

Since some well-known coding standards (e.g., JPEG, MPEG, and H.264) adopted DCT/IDCT in the compression framework, there were lots of research works emerging for the IDCT design. The majority of these researches have put efforts on optimizing the butterfly structure by minimizing the number of required multiplications and additions [4]–[6]. These strategies were widely used in hardware implements. Taking advantage of DCT coefficient redundancy is another way to reduce the complexity of IDCT process [7]–[9]. Choi *et al.* [7] utilized the characteristics of zero coefficients and replaced massive multiplications with table look-ups on MPEG-4. Chen *et al.* [8] skipped the calculations of zero coefficients on MPEG-2 and H.264/AVC by exploiting the end-of-block point and corner coefficients. Most of these works have been done for video decoding on embedded systems [10], [11].

Transform blocks with variable sizes were adopted in AVS2. Benefiting from the high intra/inter prediction efficiency of AVS2, the amplitude of prediction residuals is much smaller than that of previous standards. After transform on these residuals (especially for large transform blocks), most of the coefficients in a block are small enough to be quantized to zeroes, which are unnecessary to be processed in the following inverse transform module. Therefore, skipping the calculations of zeroes is an effective way to accelerate inverse transform process. Based on this idea, in this paper, we design several patterns for transform blocks with different sizes to detect the distribution of non-zero coefficients. If a transform block conforms to one of our designed patterns, the corresponding simplified IDCT function will be executed. Experimental results showed that our strategy could reduce the computation time by 19.3% on average under various test conditions. Moreover, our method will not result in any coding performance loss. The proposed method can be applied in both encoder and decoder.

Thanks to National Science Foundation of China 61370115, 61402018, China 863 project of 2015AA015905, and Shenzhen Peacock Plan and Fundamental Research Project for funding.

978-1-5090-2179-6/16 $31.00 © 2016 IEEE
DOI 10.1109/BigMM.2016.60

The rest of the paper is organized as follows. The conventional IDCT design and transform cores in AVS2 are briefly presented in Section II. Section III analyzes the distribution of quantized DCT coefficients and proposes our method to accelerate the IDCT procedure. Experimental results are shown in Section IV. Finally, this paper is concluded in Section V.

II. Background

A. Conventional Fast IDCT Algorithm

In this section, we review the basic principle of the conventional fast IDCT algorithm. Generally, the N-point 1-D IDCT is defined as,

$$X_{N\times 1} = D^T_{N\times N} \times Y_{N\times 1} \tag{1}$$

and N-point 2-D IDCT is formulated as,

$$X_{N\times N} = D^T_{N\times N} \times Y_{N\times N} \times D_{N\times N} \tag{2}$$

where $Y_{N\times N}$ and $X_{N\times N}$ are coefficients in the frequency and spatial domain respectively. Additionally, $D^T_{N\times N}$ and $D_{N\times N}$ are transform matrixes that convert $Y_{N\times N}$ into $X_{N\times N}$. In previous standards, N is a specific value. For example, N equals 4 or 8 in H.264. In AVS2, N can be $4, 8, 16, 32$ or 64. Moreover, the size of a block is no longer limited to $N \times N$.

$D^T_{N\times N}$ is the transpose of transform matrix $D_{N\times N}$, so the N-point 2-D IDCT can be fast computed in two steps by successive 1-D operations on the rows and columns of a block. This property is known as separability. In AVS2, the 2-D IDCT is implemented by performing 1-D operation on each row followed the 1-D operation on each column as Equation (3) indicates.

$$\begin{aligned} X_{N\times N} &= D^T_{N\times N} \times Y_{N\times N} \times D_{N\times N} \\ &= (D^T_{N\times N} \times (D^T_{N\times N} \times Y_{N\times N})^T)^T \end{aligned} \tag{3}$$

Fig.1 shows the classic 8-point 1-D IDCT signal flow graph proposed by Chen [12], which is a typical butterfly graph of the 1-D IDCT. The butterfly structure is very useful to build fast pipeline-based IDCT algorithm.

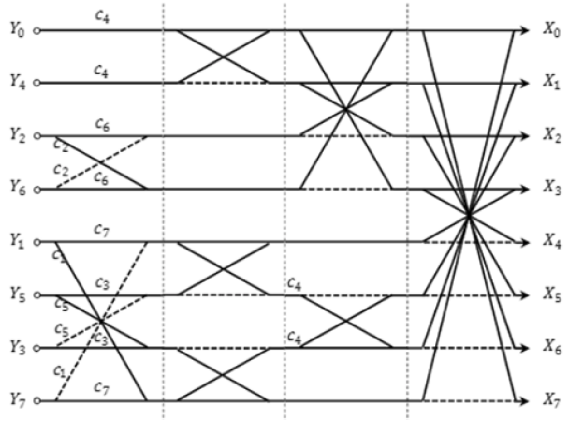

Fig. 1. 1-D 8-Point Chen IDCT butterfly graph,where $c_n = cos\frac{n\pi}{16}$

Since the matrix of DCT contains real coefficients presented by a finite number of bits, that inevitably leads to the possibility of drift (mismatch between the decoded data in the encoder and decoder). In order to eliminate the drift, the Integer Cosine Transform (ICT) is used as an approximation to DCT, in the latest video standards like H.264, HEVC and AVS2. Usually integer cosine transform must meet some restrictions, but it follows the properties of DCT/IDCT as well [13].

B. Transform Cores in AVS2

As it is mentioned above, several new transform cores were adopted in AVS2. Above all, there are a series of new sizes for transform units. The size of symmetric TUs is ranged from 4×4 to 64×64. Furthermore, the asymmetric TUs include six different sizes: $16 \times 4, 32 \times 8, 64 \times 16, 4 \times 16, 8 \times 32$ and 16×64. As for transform techniques, there are discrete cosine transform, wavelet transform (WT) and secondary transform (ST) in AVS2 codec. DCT/IDCT remains playing a major role in transform and inverse transform process.

The specific transform techniques for each size of TU are listed in TABLE I. It should be pointed out that 64×64 blocks are supposed to do wavelet transform before DCT. Thus, in their inverse transform, they ought to do IDCT first and then inverse wavelet transform. Since there are a few occurrences of IDCT process in $4 \times 4, 16 \times 64$ and 64×16 blocks, our optimizing work mainly focuses on transform blocks with other sizes.

III. Proposed fast IDCT scheme

A. Satistical Analysis and Pattern Design

As it is mentioned above, skipping the IDCT calculations of zero coefficients is a practical way to speed up the inverse transform process. Since all-zero blocks have been supposed to bypass the inverse quantization and inverse transform in AVS2, all-zero blocks are not taken into account in our algorithm.

We conduct a test to measure the probability of the non-zero coefficients in a block before IDCT process on several typical sequences with different QPs. Fig.2 shows the proportion of 16×16 blocks with different amount of non-zero coefficients on $BasketballDrill$ with $QP = 34$. In theory, there are 256 non-zero coefficients at most in a 16×16 block. However, we notice that blocks with only one non-zero coefficient occupy the largest part and the second place is taken up by blocks with 16 non-zero coefficients. Experimental data manifest that

TABLE I
TRANSFORM TECHNIQUES FOR EACH SIZE OF TU IN AVS2

TU	Transform Technique	TU	Transform Technique
4×4	ST or DCT	4×16	DCT
8×8	DCT	8×32	DCT
16×16	DCT	16×64	first WT,then DCT
32×32	DCT	16×4	DCT
64×64	first WT,then DCT	32×8	DCT
		64×16	first WT,then DCT

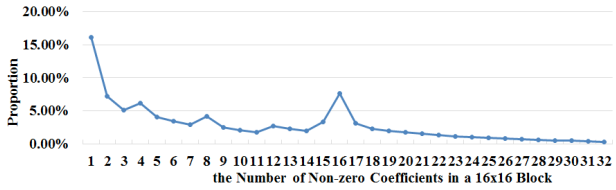

Fig. 2. Proportion of 16×16 blocks with different amount of non-zero coefficients on $BasketballDrill$ with $QP = 34$

the proportion of 16×16 blocks with 16 and less non-zero coefficients is up to 73.79% in total. That means, there is a great deal of coefficient redundancy to be cut down.

Although the distribution of non-zero coefficients cannot be derived from the number of non-zero coefficients, we can set several patterns to check the distribution according to the experimental data above. Considering that the non-zero coefficients after DCT are gathered at the upper-left corner in a block, a couple of designed patterns for 16×16 blocks are shown in Fig.3, where only the black region contains the non-zero coefficients and requires IDCT calculations. Although the rectangle-shaped design may not maximize the reduction of redundant calculations, it is convenient for simplified IDCT functions designing as well as further parallel computing optimization.

If the non-zero coefficients in a block are within the scope of the black region in P1 (in Fig.3), we define that the distribution of this block conforms to P1. Notably, P2 or P3 doesn't include the cases of P1 in our design. Tests have been run to measure the probability of each designed pattern. From TABLE II. , it indicates that the probability for P1 and P4 reaches 50% and more. Results on other sequences with different QPs also accord with this rule. In consideration of the tradeoff between patterns detection cost and accelerating ability, we choose P1 and P4 in Fig.3 as the fast patterns for 16×16 blocks.

B. Proposed Scheme

Following the methodology of pattern design for 16×16 blocks, we decide the patterns for different-sized blocks as fast IDCT modes. Fig.4 gives the designed patterns and their applied cases. For example, Mode 2 (located at the upper-right corner in Fig.4) is suitable for 16×16 and 32×32 blocks. In this figure, the black part represents the area of non-zero coefficients and the white part stands for the zeroes. Moreover, the corresponding simplified IDCT functions are provided, which only execute the IDCT calculations for the black region according to the given pattern.

TABLE II
PROBABILITY OF EACH DESINED PATTERN FOR 16×16 BLOCKS ON $BASKETBALLDRILL$ WITH $QP = 34$

Pattern	Probability	Pattern	Probability
P1	84.9%	P4	59.2%
P2	8%	P5	8.5%
P3	6.6%	P6	8%

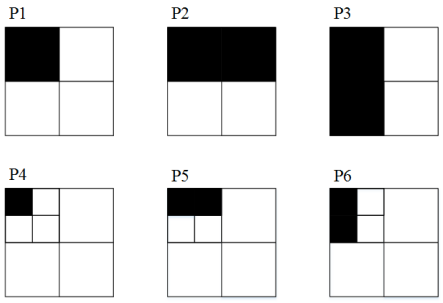

Fig. 3. Designed patterns for 16×16 blocks

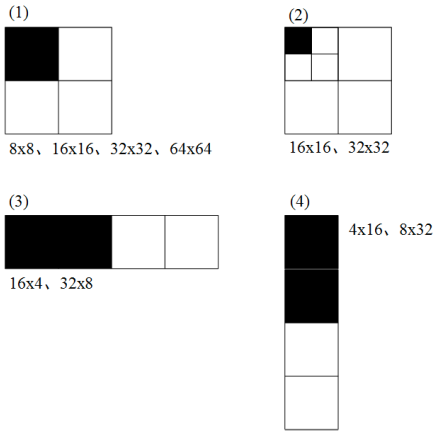

Fig. 4. Selected fast patterns and their applied cases

The mode determination for a 16x16 block is illustrated as an instance in Fig.5, where the grey part indicates non-zero coefficients in this block. For blocks of size 16×16, there are three alternative modes — two fast modes (Mode (1) and (2) in Fig.5) and a normal one (Mode (3) in Fig.5). We specify that, the more efficient mode has a higher priority to be chosen (the performance of each mode is analyzed in Section C). The distribution of non-zero coefficients in this figure doesn't match to Mode (1), and then it's detected to be in accord with mode (2). So Mode (2) is determined to be the IDCT mode for this block. The corresponding simplified IDCT function which only calculates the black part in Mode (2) will be executed. Additionally, supposing that a block doesn't meet the criteria of any fast mode, the normal mode (original IDCT in AVS2) will be carried out.

Our proposed method is demonstrated in Fig.6, which starts from inverse quantization (IQ) process and ends at IDCT functions. Since the IQ process won't change the value of zero coefficients, the first stage, obtaining the coefficients of a $M \times N$ block (M is the width of this block, N is its height), can be performed during IQ process. Next, we record the locations of non-zero coefficients while scanning and then can easily determine which computing mode it conforms to. Finally, the corresponding inverse transform function is executed.

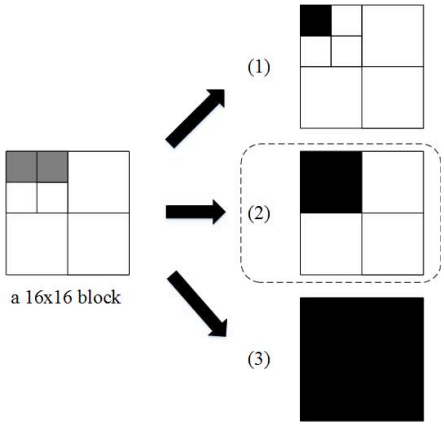

Fig. 5. An example for mode determination

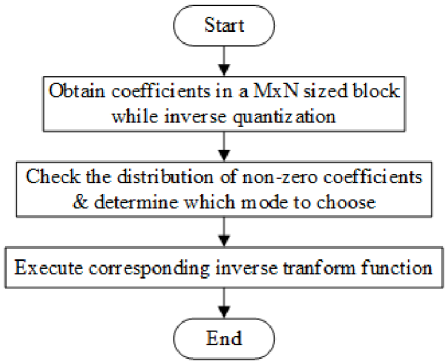

Fig. 6. Flowchart of proposed method

C. Performance Analysis

The sketch of IDCT process for a fast mode (Mode 2 in Fig.4) is expressed in Figure.7. There are only $1/16$ of total coefficients at the upper-left sub-block required to be computed in the input block (a), and (b) indicates the distribution of non-zero coefficients in this block after the first 1-D IDCT. Obviously, $15/16$ and $3/4$ of calculations will be bypassed in the first and second 1-D IDCT respectively. Therefore, as to Mode 2, the reduction of IDCT calculations can achieve 84.375% in all. In the same way, it is inferred that Mode 1 will bring the reduction by 62.5%, and Mode 3 and Mode 4 can skip about 25% calculations.

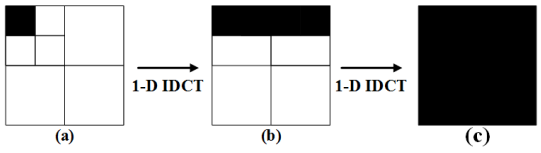

Fig. 7. IDCT process for a fast mode

IV. Experiments

A. Experiment Setup

To evaluate the performance, we implement the fast IDCT design on RD12.0 (the latest released version for AVS2) as the tested version while the original RD12.0 is the anchor. The test platform is Intel Xeon CPU E5-2670 2.50 GHz with 2 cores, 32.0 GB RAM. The resolution of test sequences is ranged from WQVGA (416×240) to UUHD (3840×2160), covering various classes. The experiments are conducted under two conditions, one is Low delay with P slices (LDP) and the other is Random Access with B slices (RA). The detailed parameters are listed in TABLE III.

B. Experimental Results

Since our proposed algorithm leads to lossless coding performance, that is the Bitrate and PSNR keep unchanged, we mainly discuss the execution time in the experimental results. Our proposed design involves two tasks: inverse quantization and inverse transform. For a fair comparison, T designates the time of inverse quantization and inverse transform in the following expressions. The time saving (TS) is defined as Equation (4), where T_0 and T_1 stands for that time in original RD12.0 and our proposal respectively.

$$TS = \frac{1}{4}\sum_{i=1}^{4}\frac{T_0(QP_i) - T_1(QP_i)}{T_0(QP_i)} \tag{4}$$

It is observed from TABLE IV. that the effect of acceleration differs from sequence to sequence and the execution time can be lowered by 19.52% and 19.09% on average under LDP and RA configurations respectively. Notably, the time saving rate can reach 23.55% in Class UUHD and UHD. As Fig.8 indicates, the larger the QP is, the higher the time saving rate will be. As the value of QP becomes larger, the coding quality is supposed to be poorer and the proportion of blocks satisfying our requirements is growing, so that the time saving rate tends to be rising in the meanwhile.

V. Conclusion

In this paper, we proposed a lossless and fast IDCT design for AVS2 by detecting the distribution of non-zero coefficients. Several patterns and corresponding fast IDCT functions are designed for transform blocks with different sizes. In our proposed method, the examination of non-zero coefficients is merged in inverse quantization process, which makes our

TABLE III
PARAMETERS FOR DIFFERENT TEST CONDITIONS IN AVS2

Parameter	LDP	RA
QP_{IFrame}	$27, 32, 38, 45$	
QP_{PFrame}	$QP_{IFrame}+1$	
QP_{BFrame}	-	$QP_{IFrame}+4$
$SeqHeaderPeriod$	0	1
$IntraPeriod$	0	8
$NumberBFrames$	0	7

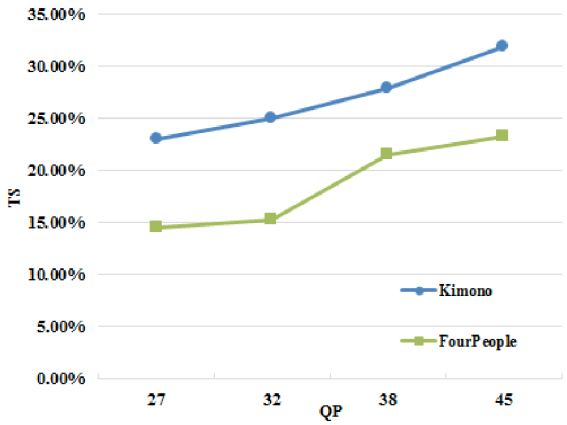

Fig. 8. Time saving perfomance under different values of QP

TABLE IV
TIME SAVING PERFORMANCE COMPARISON BEWTEEN RD12.0 AND PROPOSED METHOD

Resolution	Sequence	TS		
		LDP	RA	Average
UUHD & UHD	pku_girls	25.05%	24.87%	23.55%
	pku_parkwalk	25.18%	25.46%	
	Traffic	19.33%	21.39%	
1080p	beach	21.66%	21.45%	21.27%
	taishan	15.54%	16.44%	
	Kimono	27.08%	27.02%	
	Cactus	21.95%	20.15%	
	BasketballDrive	20.94%	20.45%	
WVGA	BasketballDrill	23.40%	21.94%	17.81%
	BQMall	17.25%	17.18%	
	PartyScene	12.01%	13.48%	
	RaceHorses	18.81%	18.37%	
WQVGA	BasketballPass	17.92%	18.04%	14.85%
	BlowingBubbles	14.29%	14.61%	
	BQSueare	10.41%	9.48%	
	RaceHorses	16.84%	17.18%	
720p	City	13.39%	13.80%	19.52%
	Crew	24.68%	24.29%	
	Vidyo1	22.57%	20.63%	
	Vidyo3	19.62%	17.94%	
	FourPeople	22.01%	18.67%	
	Johnny	19.50%	17.14%	
Average		**19.52%**	**19.09%**	**19.30%**

algorithm available in both encoder and decoder. Experimental results showed that our method could achieve about 19.3% time reduction without any performance degradation.

REFERENCES

[1] AVS Working Group Website: http://www.avs.org.cn.
[2] S. Ma, T. Huang, and W. Gao, *The second generation IEEE 1857 video coding standard*, Signal and Information Processing (ChinaSIP), 2015 IEEE China Summit and International Conference on. IEEE, 2015.
[3] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, *Overview of the high efficiency video coding (hevc) standard*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649 ?1668, 2012.
[4] C. Loeffler, A. Ligtenberg, and G. S. Moshytz, *Practical fast 1-D DCT algorithms with 11 multiplications*, in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, vol. 2, pp. 988-991, May 1989.
[5] A. Saxena, F. C. Fernandes, and Y. Reznik, *Fast transforms for intra-prediction-based image and video coding*, Data Compression Conference (DCC), 2013. IEEE, 2013.
[6] T. Ma, C. Liu, Y. Fan and X. Zeng, *A fast 88 IDCT algorithm for HEVC*, ASIC (ASICON), 2013 IEEE 10th International Conference on. IEEE, 2013.
[7] K. Choi, S. Lee, and E. S. Jang, *Zero coefficient-aware IDCT algorithm for fast video decoding*, IEEE Trans. Consum. Electron., vol. 56, no. 3, pp. 1822-1829, Aug 2010.
[8] O. T. C. Chen, M. L. Hsia, and C. C. Chen, *Low-complexity inverse transforms of video codecs in an embedded programmable platform*, Multimedia, IEEE Transactions on 13.5 (2011): 905-921.
[9] K. Lee, K. Choi, and E. S. Jang, *Fixed-point zero coefficient-aware fast IQ-IDCT algorithm*, Consumer Electronics-Berlin (ICCE-Berlin), 2011 IEEE International Conference on. IEEE, 2011.
[10] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze and M. Sadafale, *Core transform design in the high efficiency video coding (HEVC) standard*, Selected Topics in Signal Processing, IEEE Journal of 7.6 (2013): 1029-1041.
[11] Z. Jia, Z. Liu, and D. Wang, *Fully pipelined DCT/IDCT/Hadamard unified transform architecture for HEVC Codec*, Circuits and Systems (ISCAS), 2013 IEEE International Symposium on. IEEE, 2013.
[12] W. H. Chen, C. Smith and S. Fralick, *A Fast Computational Algorithm for the Discrete Cosine Transform*, IEEE Transactions on Communications, vol.25, no.9, pp. 1004- 1009, Sep 1977.
[13] W. K. Cham, *Development of integer cosine transforms by the principle of dyadic symmetry*, IEEE Proceedings I, Communications, Speech and Vision, vol.136, no.4, pp. 276- 282, Aug 1989.