HIGH-ACCURACY AND LOW-COMPLEXITY FIXED-POINT INVERSE DISCRETE COSINE TRANSFORM BASED ON AAN'S FAST ALGORTIHM

Honggang Qi and Wen Gao
Institute of Computing Technology, Chinese Academy of Sciences
E-mail: {hgqi, wgao}@jdl.ac.cn

Abstract—In this paper, a high accuracy and low-complexity fixed-point inverse discrete cosine transform (IDCT) based on AAN's fast algorithm is proposed for the approximation of the real IDCT. The proposed transform can be implemented with 16-bit multiplication and 24-bit addition operations. A common factor extraction algorithm is used to obtain a new set of real factors of IDCT whose integer approximations can be implemented with few arithmetic operations. A two-stage scale approach is proposed for constraining the dynamic range to a narrow range in scale part. The experimental results show that the proposed transform has significantly higher accuracy than the thresholds of IEEE1180-1190. The results of the proposed IDCT implemented into the decoder of MPEG-2 also show that the fixed-point IDCT achieves higher performance.

Index Terms — Fixed-point IDCT, AAN's fast algorithm, IEEE1180-1190.

1. INTRODUCTION

Discrete Cosine Transform (DCT) is used in video or image coding for decorrelating the spatial signals of pictures. It converts the spatial signals to the sparse transformed signals so that the transformed signals can be more efficiently compressed in following entropy coding. The inverse DCT (IDCT) is the inverse process of the DCT. It is used to reconstruct the spatial signals from the transformed signals. The ideal DCT/IDCT has the perfect reconstruction property. Theoretically, the ideal DCT/IDCT is defined as real number operations. Since the limit of signal processing technology, many early video or image coding standards (MPEG-1, MPEG-2, MPEG-4 part 2, H.263 and JPEG) adopted the floating-point 8x8 DCT/IDCT to compress the videos or images directly. However, the floating-point DCT/IDCT implementation is high-complexity, thus these standards allow the individual decoders to approximate the floating-point 8x8 IDCT in practical applications. The fixed-point IDCT in decoder reduces the complexity of transform while involving in the error drift problem. Error drift problem is caused by the inexact decoded pictures between encoder and decoder. When these inexact decoded pictures are referred by inter-coding pictures, the errors may be accumulated. When there are no intra-refresh pictures in a long sequence of inter-coding pictures, the increasing error accumulation may worsen the qualities of late decoded pictures severely, especially in quarter-sample precision motion compensation used in inter-prediction of decoder. The degree of error drift is dependent on the accuracy of fixed-point IDCT. The higher accuracy is, the smaller error drift is. Thus, it is necessary to design a high-accuracy fixed-point IDCT to reduce the error drifts of decoders. On dealing with error drift, the drift-free is our final goal, thus, accuracy is first considered in our fixed-point IDCT implementation. Under the precondition that small error drift can not worsen the decoded pictures, the complexity should be reduced as much as possible. Therefore, a trade-off between accuracy and complexity should be done in the implementation of fixed-point IDCT.

Directly implementation of IDCT according to the IDCT definition needs a large number of arithmetic operations, thus many fast DCT/IDCT algorithms are proposed to accelerate DCT/IDCT, such as Chen's [1], Loeffler's [2] and AAN's [3] fast algorithm. All these fast algorithms reduce the number of arithmetic operations greatly, thus are widely applied to the implementations of fixed-point IDCTs in decoders. Although the number operations of floating-point IDCT can be reduced based on these fast algorithms, the 64-bit floating-point operations are the other obstacle for its implementation. The fixed-point IDCT applied in decoder avoids the complicated floating-point operations. But, the high-accuracy integer IDCT is usually achieved at the cost of more complicated integer factors and wider dynamic range which increases the complexity of fixed-point IDCT. In this paper, a high-accuracy fixed-point 8x8 IDCT based on AAN's fast algorithm is proposed for reducing the error drifts and complexities of decoders. AAN's IDCT is a scale transform which consists two parts, one is scale, and the other is co-transform. In the scale part, a two-stage scale approach is proposed to divide a scale factor with wide bit width into two scale factors with narrow bit widths so that the high-accuracy scale is implemented in narrow bit widths. In the co-transform part, the common factor extraction algorithm is used to obtain a set of real factors so that the integer factors approximated from these real factors are implemented with fewer arithmetic operations.

The rest of this paper is organized as follows, in the section 2, the basic architecture on which the proposed fixed-point IDCT based is first presented, and then the common factor extraction algorithm and the two-stage approach are described. In the section 3, the experimental results are shown. Finally, this paper is concluded in the section 4.

2. THE DESIGN OF FIXED-POINT 8X8 IDCT

2.1. The architecture of fixed-point IDCT based on AAN's fast algorithm

AAN's fast IDCT algorithm is a typical architecture which is widely adopted in low-complexity implementations. It is a scale transform. The merit of scale transform is that the complicated factors inside transform are removed to the scale part as the scale factors. As a result, the complexity of transform will be reduced. A main characteristic of AAN's fast IDCT is that only 4 different factors are left inside transform for necessary computation, and the implementation of these factors needs 5 multiplications and 29 additions totally. There is one multiplication at most in each signal path which is effective for

parallel implementation [3]. The butterfly structure of AAN's fast 8x8 IDCT with a small modification [4] is presented in Fig.1. The difference in the modified AAN's IDCT is that 3 multiplications in the odd part of the AAN's IDCT are replaced by a simple plane rotation. The modified AAN's IDCT has 3 different factors and needs 6 multiplications and 28 additions in core-transform. The modified AAN's IDCT has fewer data dependence than the original AAN's IDCT. Thus, it is easier for implementation. The 8x8 input data are first scaled in the scale part. These data are usually scaled in 2-D space with an 8x8 scale factors. After scale, the outputs of 8x8 scaled data are processed row-by-row then colum-by-colum in the core-transform.

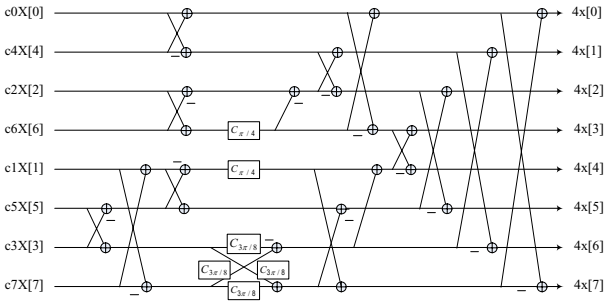


Fig.1. 1-D IDCT butterfly architecture of modified AAN's fast algorithm.

In the proposed fixed-point implementation of IDCT, the real factors are usually approximated in the form of an integer numerator divided by an integer denominator equivalent to the power of 2, i.e. $f \approx D/2^n$, where f is the real factor, D is the integer numerator and n is the non-negative integer. The form of $D/2^n$ as the integer approximation of real factor is used as the integer factors of fixed-point IDCT. In this way, the real multiplication can be approximately replaced with integer multiplication and right-shift. According to $f \approx D/2^n$, the integer numerator D can be obtained through multiplying the real factor f by 2^n, and then rounding to the nearest integer, i.e. $D = round(2^n f)$. The approximation accuracy is depended on the value of n. The approximation is more accurate with the increment of n, while the dynamic range is also widened. Thus, the selection of n is a trade-off between accuracy and complexity.

2.2. Common factor extraction algorithm

A common factor extraction algorithm is used to obtain an optimal combination of factors for achieving a high-accuracy and low-complexity fixed-point IDCT. The 1-D core-transform is composed of even part and odd part. Two different factors λ and τ are respectively used as the common factors of even part and odd part. With the common factor extraction, the new factor pairs are obtained as follows

$$\begin{aligned}(\alpha 1, \beta 1, \theta, \varphi) &= (1, \cos\pi/4, \cos 3\pi/4, \sin 3\pi/4) / \tau, \quad \text{odd part} \\ (\alpha 2, \beta 2) &= (1, \cos\pi/4) / \lambda, \qquad\qquad \text{even part}\end{aligned} \tag{1}$$

The improved AAN's IDCT is illustrated in Fig.2. The scale factor C_i in the improved IDCT is equivalent to the c_i multiplied by corresponding common factor λ or τ.

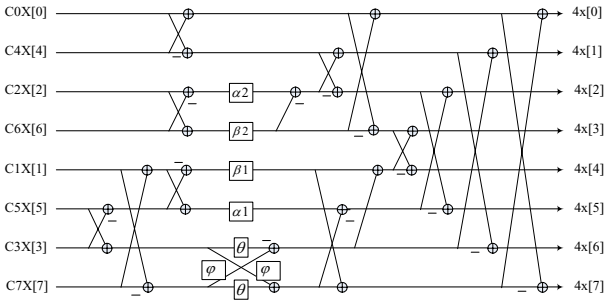


Fig.2. 1D-IDCT factorization of modified AAN's fast algorithm.

The different common factor pairs of λ and τ will cause different transform results and complexities of fixed-point IDCT implementations. Through adjusting the values of λ and τ, the IDCT with different accuracies and complexities can be obtained. In this paper, proper values of λ and τ are selected for the proposed fixed-point IDCT. According to the selected values of λ and τ, a new set of real factors of IDCT are determined from (1). With the integer factors approximated from the new set of real factors, the proposed fixed-point IDCT can achieve high accuracy and low complexity simultaneously. The integer approximations of factors in proposed fixed-point IDCT are shown in Table 1.

Table 1. Integer approximations of real factors

REAL FACTORS IN IMPROVED IDCT		*INTEGER APPROXIMATION*
$\alpha 1$	$1/\tau$	17474/16384
$\beta 1$	$\cos(\pi/4)/\tau$	12356/16384
θ	$\cos(3\pi/4)/\tau$	6687/16384
φ	$\sin(3\pi/4)/\tau$	16144/16384
$\alpha 2$	$1/\lambda$	17378/16384
$\beta 2$	$\cos(\pi/4)/\lambda$	12288/16384

2.3. Two-stage scale approach

Usually, the scale is executed once before the 2-D core-transform. The 2-D scale factor C_{ij} is the element of scale matrix S in position (i, j). C_{ij} is computed from C_iC_j. The integer scale factors S in the proposed IDCT are the integer rounding of 2^{17} times of real factors C_{ij}, i.e. $S = round(2^{17} C_{ij})$. According to it, the 64 integer scale factors in scale matrix S are obtained and presented as

$$S = \begin{bmatrix} 16384 & 19552 & 28542 & 11075 & 16384 & 55679 & 11823 & 13064 \\ 19552 & 23333 & 34061 & 13217 & 19552 & 66446 & 14108 & 15590 \\ 28542 & 34061 & 49723 & 19294 & 28542 & 96998 & 20596 & 22759 \\ 11075 & 13217 & 19294 & 7487 & 11075 & 37638 & 7992 & 8831 \\ 16384 & 19552 & 28542 & 11075 & 16384 & 55679 & 11823 & 13064 \\ 55679 & 66446 & 96998 & 37638 & 55679 & 189221 & 40178 & 44397 \\ 11823 & 14108 & 20596 & 7992 & 11823 & 40178 & 8531 & 9427 \\ 13064 & 15590 & 22759 & 8831 & 13064 & 44397 & 9427 & 10417 \end{bmatrix} \tag{2}$$

For guaranteeing the intermediate results of transform in 24-bit in following core-transform, the scaled input data are down-scaled 7-bit. The process of scale is described as

$$(F[i][j]*S[i][j]) >>7 \tag{3}$$

where $F[i][j]$ denotes the input data in the position (i, j), and $S[i][j]$ denotes the integer scale factor in the corresponding position of scale matrixes S. The maximum integer factor in S is 18-bit unsigned integer. Thus, it needs 18-bit unsigned multiplications in the scale, which is the complicated operation in both the hardware and the DSP implementations. For solving this problem, a two-stage scale approach is proposed for constraining the multiplications no more than 16-bit. This approach splits an 18-bit multiplication into two less than 16-bit multiplications. The process of two-stage scale is expressed as

$$(F[i][j]*S1[i][j]) + (F[i][j]*S2[i][j]>>7) \tag{4}$$

where $S1[i][j]$ and $S2[i][j]$ denote the two scale factors in position (i, j). $S1$ is the 11-bit unsigned integer scale matrix and $S2$ is the 7-bit signed integer scale matrix. The two scale matrixes are expressed as follows:

$$S1 = \begin{bmatrix} 128 & 153 & 223 & 87 & 128 & 435 & 92 & 102 \\ 153 & 182 & 266 & 103 & 153 & 519 & 110 & 122 \\ 223 & 266 & 388 & 151 & 223 & 758 & 161 & 178 \\ 87 & 103 & 151 & 58 & 87 & 294 & 62 & 69 \\ 128 & 153 & 223 & 87 & 128 & 435 & 92 & 102 \\ 435 & 519 & 758 & 294 & 435 & 1478 & 314 & 347 \\ 92 & 110 & 161 & 62 & 92 & 314 & 67 & 74 \\ 102 & 122 & 178 & 69 & 102 & 347 & 74 & 81 \end{bmatrix} \tag{5}$$

$$S2 = \begin{bmatrix} 0 & -32 & -2 & -61 & 0 & -1 & 47 & 8 \\ -32 & 37 & 13 & 33 & -32 & 14 & 28 & -26 \\ -2 & 13 & 59 & -34 & -2 & -26 & -12 & -25 \\ -61 & 33 & -34 & 63 & -61 & 6 & 56 & -1 \\ 0 & -32 & -2 & -61 & 0 & -1 & 47 & 8 \\ -1 & 14 & -26 & 6 & -1 & 37 & -14 & -19 \\ 47 & 28 & -12 & 56 & 47 & -14 & -45 & -45 \\ 8 & -26 & -25 & -1 & 8 & -19 & -45 & 49 \end{bmatrix} \tag{6}$$

3. COMPLXITY ANALYSIS

The multiplication-free operations of the integer factors in the core-transform of the proposed IDCT and their complexities are shown in Table 2. It needs to be explained that intermediate computing results of 6687/16384 can be reused in the computations of 16144/16384, thus the total complexity of 6687/16384 and 16144/16384 is shown in Table 2. According to the Table 2, the complexity of 1-D core-transform is 28 + 3 + 3 + 3 + 1 + 2 x 4 = 46 additions and 2 + 3 + 2 + 1 + 2 x 5 = 18 shifts, and the complexity of 2-D core-transform are 16 x 46 = 736 additions and 16 x 18 = 288 shifts. Compared with the fast integer transform in MPEG-2 TM5 [5] which need 11 multiplications and 29 additions in 1-D transform, the complexity of the multiplication-free implementation of the proposed IDCT is lower.

The scale factors can be absorbed in the inverse quantization, which can not increase any complexity to inverse quantization. The scale may also be independently implemented through multiplying input data with corresponding scale factors. It needs to be performed only for non-zero coefficients. Therefore instead of executing 64 multiplications during the scaling stage, in a typical video decoding scenario it would be sufficient to execute only K multiplications which can be as small as 4 or 5 [6]. Thus, the complexity of scale is $2K$ multiplications (K<=5). Since the integer factors in both core-transform and scale are properly scaled, the dynamic range in the proposed IDCT is constrained in 24-bit width. The bit width of the proposed IDCT is narrower than the 32-bit width of fast integer IDCT in TM5.

Table 2. Integer factors' multiplication-free implementation and their complexities.

Integer approximation	*Multiplication-free operations*	*Complexity*	
		addition	*shift*
17474/16384	x1 = i + (i>> 4), x2 = -i – x1, o = x1 – (x2 >> 9);	3	2
12356/16384	x1 = i – (i >> 2), x2 = i – (i >> 4), o = x1 + (x2 >> 8);	3	3
6687/16384	x1 = (i>> 5), x2 = (i >> 1) – x1, o = i – (x2 >> 5);	4	5
16144/16384	x3 = x2 – (o >> 4), o = x3 + (x1>>5);		
17378/16384	x1 = (i >> 9), x2 = i + x1, x3 = i + (x2 >> 4), o = x3 – x1;	3	2
12288/16384	o = i – (i >> 2);	1	1

4. EXPERIMENTAL RESULTS

The proposed 8x8 fixed-point IDCT is tested on the requirements of IEEE1180-1190 [7]. The 100,000 and 1,000,000 8x8 data blocks in five different integer intervals [-5, 5], [-256, 255], [-300, 300], [-384, 383], [-512, 511] are input into both 64-bit floating-point IDCT and proposed fixed-point IDCT. And their outputs are compared. The peak mean square errors (pmse), overall mean square errors (omse), peak mean errors (pme) and overall mean errors (ome) are used as the accuracy metrics to measure the accuracy of the proposed fixed-point IDCT. The test results in the metrics of IEEE1180-1190 are shown in Table 3. It is observed that the proposed IDCT achieves 10 times more than the thresholds of IEEE1180-1190 in omse. Moreover, the MPEG-2 TM5 is used as the test bench to test the practical error drift of decoder. The encoder of TM5 uses the 64-bit floating-point DCT and IDCT, and the proposed fixed-point IDCT is implemented into the decoder of TM5. Two typical sequences *foreman* and *mobile* in CIF format are tested. 300 frames of each sequence are coded into a group of pictures (GOP). In the GOP, just the first picture is coded as intra picture and left pictures are

coded as inter pictures. The Quantization Scale (QS) is fixed to 1 for testing the maximum error drift in the extreme condition. The drift PSNR between the reconstructed picture in encoder and the corresponding decoded picture in decoder is employed as the error drift criterion. The drift PSNRs of decoded pictures produced by the TM5 decoders with three different IDCTs are shown in Fig.3. It can be observed that the drift PSNR curves of proposed IDCT are closer to the drift PSNR curves of floating-point IDCT than these of fast integer IDCT in TM5, which indicates that fewer drifts occur between ideal floating-point IDCT and proposed fixed-point IDCT.

Table 3. Results of fixed-point IDCT on the metrics of IEEE1180-1190.

Q	L, H	S	pmse (<0.06)	omse (<0.02)	pme (<0.015)	ome (<0.0015)
10^4	5, 5	+	5.90e-003	2.35e-003	5.90e-003	1.73e-004
10^4	5, 5	-	6.00e-003	2.28e-003	6.00e-003	-7.81e-006
10^4	256, 255	+	7.00e-003	2.75e-003	7.00e-003	3.59e-005
10^4	256, 255	-	9.00e-003	2.91e-003	9.00e-003	4.06e-005
10^4	300, 300	+	5.80e-003	2.46e-003	5.80e-003	-7.81e-005
10^4	300, 300	-	7.50e-003	2.44e-003	7.50e-003	3.59e-005
10^4	384, 383	+	5.10e-003	2.13e-003	5.10e-003	3.44e-005
10^4	384, 383	-	5.30e-003	2.15e-003	5.30e-003	-6.88e-005
10^4	512, 511	+	4.40e-003	1.86e-003	4.20e-003	-1.23e-004
10^4	512, 511	-	3.70e-003	1.85e-003	3.70e-003	1.45e-004
10^6	5, 5	+	6.09e-003	2.31e-003	6.09e-003	4.17e-005
10^6	5, 5	-	6.14e-003	2.31e-003	6.14e-003	3.79e-005
10^6	256, 255	+	7.49e-003	2.80e-003	7.49e-003	-2.83e-006
10^6	256, 255	-	7.34e-003	2.80e-003	7.34e-003	2.77e-006
10^6	300, 300	+	6.37e-003	2.53e-003	6.36e-003	1.01e-005
10^6	300, 300	-	6.39e-003	2.53e-003	6.38e-003	2.66e-007
10^6	384, 383	+	4.97e-003	2.18e-003	4.96e-003	-7.03e-006
10^6	384, 383	-	5.01e-003	2.17e-003	4.99e-003	-2.88e-006
10^6	512, 511	+	3.83e-003	1.92e-003	3.77e-003	9.06e-007
10^6	512, 511	-	3.63e-003	1.92e-003	3.60e-003	3.88e-006

Q – the number of input blocks;
L, H – input data range [L, H];
S – the sign of input data.

5. CONCLUSION

In this paper, an 8x8 fixed-point IDCT based on modified AAN's fast algorithm is proposed. A set of new factors for proposed fixed-point IDCT is obtained with the common factor extraction algorithm, and the two-stage scale approach is used to constrain the multiplication operations within no more than 16-bit. The proposed fixed-point IDCT not only is easily implemented in decoder but also reduces error drifts of decoders. The experimental results show that the proposed IDCT can achieve a higher accuracy and lower complexity than the existing fast integer IDCT in TM5.

6. ACKNOWLEDGMENT

The work on this paper was carried out using some of the IDCT design methodologies suggested by Yuriy A. Reznik (Qualcomm), and Arianne T. Hinds (IBM). Their help and instruction are hereby recognized.

6. REFERENCES

[1] W. Chen, C. H. Smith and S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform", IEEE Transactions on Communications, vol. com-25, No. 9, pp. 1004-1009, September 1977.

[2] C. Loeffler, A. Ligtenberg and G. Moschytz, "Practical Fast 1-D DCT Algorithms with 11 Multiplications", International Conference on Acoustics, Speech, and Signal Processing 1989, pages 988-99, 1989.

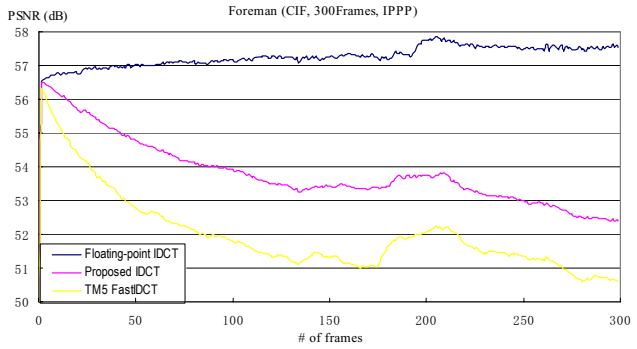
[3] Y. Arai, T.Agui and M. Nakajima, "A Fast DCT-SQ Scheme for Images", Transactions of the IEICE E71(11): 1095, November 1988.

[4] Y. A. Reznik, A. T. Hinds, N. Rijavec, Low Complexity Fixed-Point Approximation of Inverse Discrete Cosine Transform, in Proc. 32nd Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP'07), Honolulu, HI, April 15 - 20, 2007. (Accepted)

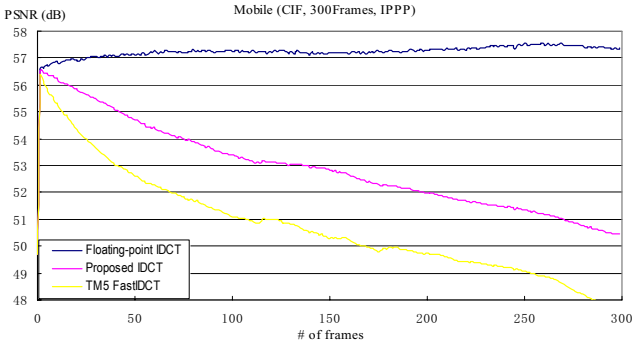
[5] MPEG-2 TM5 reference software: ftp://ftp.mpegtv.com/pub/mpeg/mssg/mpeg2vidcodec_v12.tar.gz.

[6] Arianne T. Hinds, Yuriy A. Reznik, Phoom Sagetong, Honggang Qi, Siwei Ma, Antonio Navarro, "On benefits of standardizing scaled IDCT architecture", MPEG M13311, Montreux, Switzerland, Apr. 2006.

[7] IEEE CAS Standards Committee, "IEEE Standard Specifications for the Implementations of 8x8 Inverse Discrete Cosine Transform", IEEE standard 1180-1190, Dec. 1990.



(a)



(b)

Fig.3. Drift PSNRs of pictures decoded by floating-point IDCT, proposed IDCT and TM5's fast integer IDCT.