

Efficient Large Size Transforms for High-Performance Video Coding

Rajan Joshi, Yuriy A. Reznik*, and Marta Karczewicz
Qualcomm Inc, 5775 Morehouse Drive, San Diego, CA, USA 92121

ABSTRACT

This paper describes design of transforms for extended block sizes for video coding. The proposed transforms are orthogonal integer transforms, based on a simple recursive factorization structure, and allow very compact and efficient implementations. We discuss techniques used for finding integer and scale factors in these transforms, and describe our final design. We evaluate efficiency of our proposed transforms in VCEG's H.265/JMKTA framework, and show that they achieve nearly identical performance compared to much more complex transforms in the current test model.

Keywords: discrete cosine transform, DCT, factorization, multiplicative complexity, scaled transform, video coding, H.264, H.265, HEVC.

1. INTRODUCTION

Discrete Cosine Transform of type II (DCT-II)¹⁻³ is a fundamental operation performed by the majority of today's image and video compression algorithms. It was first suggested by N. Ahmed, T. Natarajan, and K. R. Rao,¹ and subsequent research provided a number of theoretical arguments for its use, such as energy compaction property, asymptotic equivalence of DCT to the Karhunen-Loève transform for signals produced by Markov-1 process with high correlation coefficient, etc. (see, e.g. [3, Chapter 3] for a survey of the related results).

DCT-II of size 8 has served as the transform of choice in H.261, JPEG, MPEG-1, MPEG-2, H.263, and MPEG-4 visual standards.^{2,4-9} More recent standards, such as MPEG-4 AVC | H.264,¹⁰ VC-1,¹¹ and AVS¹² have adopted integer approximations of DCT-II with transform sizes: 4, 8, and 16. An emerging JPEG-XR image compression standard¹³ uses overlapping transforms, which are also based on 4-point DCT-II kernels.

A new emerging standard, High Efficiency Video Coding (HEVC), currently under development by Joint Collaborative Team of video experts from MPEG and ITU-T SG16 (JCT-VC),¹⁴ includes a number of integer transforms of sizes ranging from 4 to 64. As video resolutions keep increasing, it is possible that even larger transforms will be considered in the future.

In this paper we describe design of scaled integer transforms, which are numerically stable, fully recursive in structure, and remain orthogonal with perfect scaling (in the absence of quantization). As such, they are well suitable for use in future video coding applications. Described transforms have been proposed to ITU-T SG16 Q6 (VCEG) standardization committee,²¹ and were also included in Qualcomm's response to JCT-VC call for proposals²².

This paper is organized as follows. In Section 2, we describe design of underlying factorization that we use in the transform. Section 3 discusses conversion to integer arithmetic and other implementation aspects. Section 4 provides experimental results obtained using this transform in ITU-T SG16 Q6 JMKTA video coding model. Conclusions are drawn in Section 5.

2. FACTORIZATION

Let $\{x_n\}$, $n = 0, \dots, N-1$ be a sequence of input samples (i.e. line of pixel values). DCT-II and its inverse transform over this sequence are defined as follows:

$$X_k^{II} = \sqrt{\frac{2}{N}} \lambda(k) \sum_{n=0}^{N-1} x_n \cos\left(\frac{(2n+1)k\pi}{2N}\right), \quad k = 0, \dots, N-1,$$

* Corresponding author. Email: yreznik@ieee.org, Phone: 858-658-1866.

$$x_n = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X_k^H \lambda(k) \cos\left(\frac{(2n+1)k\pi}{2N}\right), \quad n = 0, \dots, N-1,$$

where $\lambda(k) = 1/\sqrt{2}$, if $k = 0$, and 1 otherwise. DCT-IV transform and its inverse are defined as follows:

$$X_k^{IV} = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{4N} (2n+1)(2k+1)\right), \quad k = 0, \dots, N-1,$$

$$x_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} X_k^{IV} \cos\left(\frac{\pi}{4N} (2n+1)(2k+1)\right), \quad n = 0, \dots, N-1.$$

We note that in video coding, we usually work with $N \times N$ matrices of input data, and so the above transforms need to be applied to all rows and columns in a separable fashion, to produce corresponding matrices of transform coefficients². Hereafter we will adopt such separable model in our design of integer 2D transforms, and will focus mainly on speeding up computations of the component 1D transforms[†].

For further convenience, we omit normalization factors ($\sqrt{2/N}$ and $\lambda(k)$) and define matrices:

$$C_N^H(n, k) = \cos\left(\frac{\pi(2n+1)k}{2N}\right), \quad n, k = 0, \dots, N-1,$$

$$C_N^{IV}(n, k) = \cos\left(\frac{\pi(2n+1)(2k+1)}{4N}\right), \quad n, k = 0, \dots, N-1,$$

representing coefficients of DCT-II, and DCT-IV transforms correspondingly.

It is well known, that even-sized DCT-II matrix can be factored into a product containing direct sum of smaller DCT-II and DCT-IV matrices as follows^{2,3}:

$$C_N^H = P_N \begin{pmatrix} C_{N/2}^H & \mathbf{0} \\ \mathbf{0} & C_{N/2}^{IV} J_{N/2} \end{pmatrix} \begin{pmatrix} I_{N/2} & J_{N/2} \\ J_{N/2} & -I_{N/2} \end{pmatrix}, \quad (1)$$

where P_N is a permutation matrix producing reordering:

$$x'_i = x_{2i}, \quad x'_{N/2+i} = x_{2i+1}, \quad i = 0, 1, \dots, N/2-1, \quad (2)$$

and where $I_{N/2}$ and $J_{N/2}$ denote $N/2 \times N/2$ identity and order reversal matrices respectively.

Chen-Smith-Fralick¹⁵, Wang, and many other well-known DCT-II factorizations^{2,3} rely on factorization (1) as a basic step in their decimation process.

We next apply the following decomposition to the DCT-IV block in (1):

$$C_N^{IV} = P_N^T \begin{pmatrix} 1 & & 0 \\ I_{N/2-1} & I_{N/2-1} & \\ I_{N/2-1} & -I_{N/2-1} & \\ 0 & & -1 \end{pmatrix} \begin{pmatrix} I_{N/2} & \mathbf{0} \\ \mathbf{0} & E_{N/2} J_{N/2} \end{pmatrix} \begin{pmatrix} C_{N/2}^H & \mathbf{0} \\ \mathbf{0} & C_{N/2}^{IV} \end{pmatrix} \begin{pmatrix} I_{N/2} & \mathbf{0} \\ \mathbf{0} & E_{N/2} \end{pmatrix} R_N, \quad (3)$$

where:

P_N is a reordering matrix (2), $E_{N/2}$ is the diagonal sign-alteration matrix

$$E_{N/2} = \text{diag}\{(-1)^k\}, \quad k = 0, 1, \dots, N/2-1, \quad (4)$$

[†] While faster non-separable 2D designs can possibly be created,³ their large expanded structure, and the need for support of multiple (including hybrid, e.g. 8x16) block sizes makes this approach much less appealing in practice.

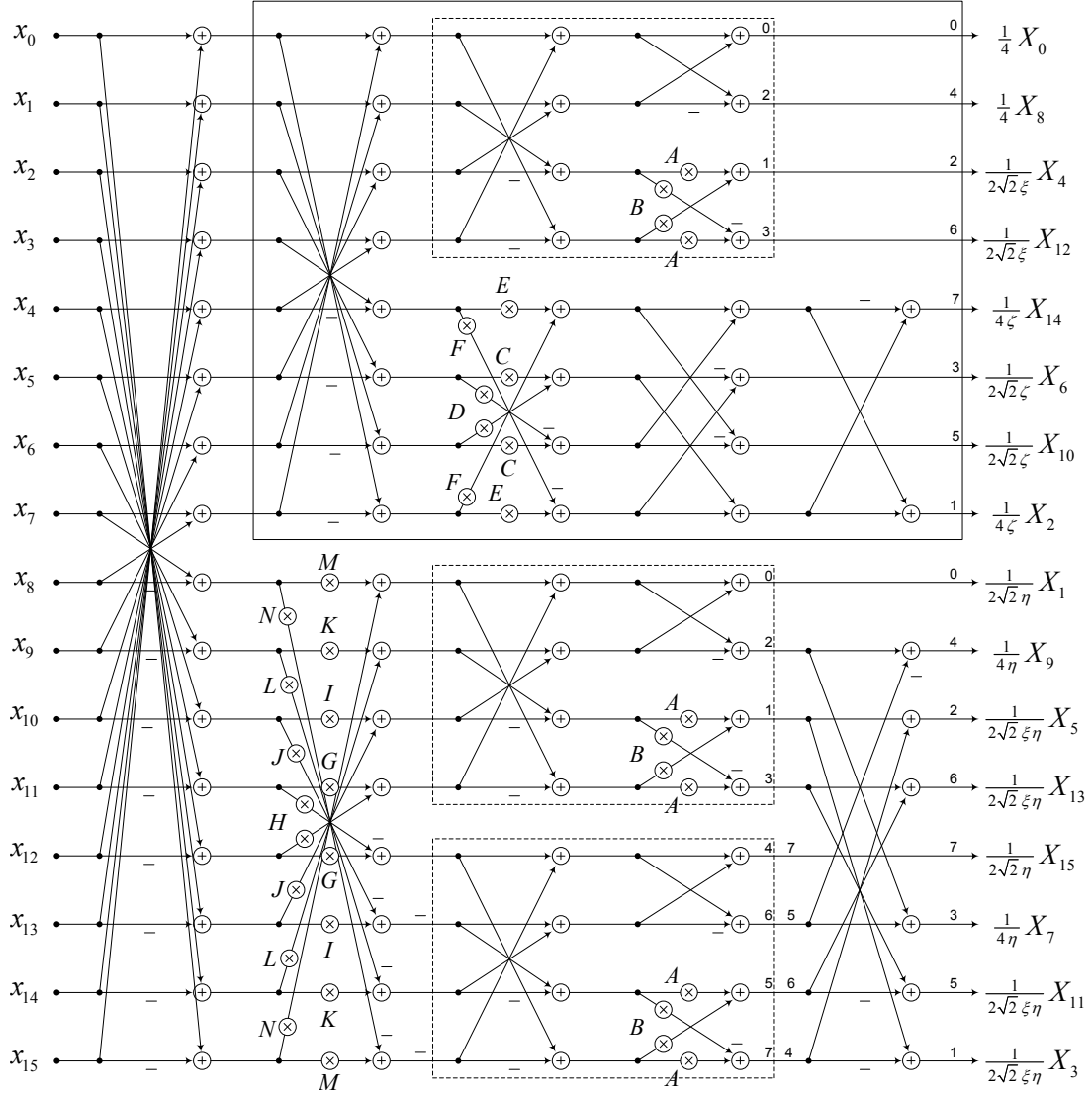


Figure 1. N=16-point DCT-II factorization produced by recursive use of decompositions (1) and (3). Note that it uses 3 identical 4-point DCT-II blocks. Scale factors ξ, ζ, η are introduced to simplify conversion of this transform to integer arithmetic. Ideally, the following equations must be achieved to make this transform a DCT-II:

$$\frac{A}{\xi} = \cos\left(\frac{3\pi}{8}\right); \frac{B}{\xi} = \sin\left(\frac{3\pi}{8}\right); \frac{C}{\zeta} = \cos\left(\frac{\pi}{16}\right); \frac{C}{\zeta} = \sin\left(\frac{\pi}{16}\right); \frac{E}{\zeta} = \cos\left(\frac{3\pi}{16}\right); \frac{F}{\zeta} = \sin\left(\frac{3\pi}{16}\right); \frac{G}{\eta} = \cos\left(\frac{7\pi}{32}\right); \frac{H}{\eta} = \sin\left(\frac{7\pi}{32}\right);$$

$$\frac{L}{\eta} = \cos\left(\frac{5\pi}{32}\right); \frac{J}{\eta} = \sin\left(\frac{5\pi}{32}\right); \frac{K}{\eta} = \cos\left(\frac{3\pi}{32}\right); \frac{L}{\eta} = \sin\left(\frac{3\pi}{32}\right); \frac{M}{\eta} = \cos\left(\frac{\pi}{32}\right); \frac{N}{\eta} = \sin\left(\frac{\pi}{32}\right); .$$

Hence, considering all above, the task of design of an integer approximation of DCT-II now boils down to finding some small integers A-N, such that conditions (7) are met, and such that the resulting fractions produce good approximations of sine and cosine factors in the transform (6).

One can easily solve this task by straightforward enumeration. As an example, we show several example solutions found for factors A and B (and associated scale factor ξ) in Table 1.

Table 1. Example factors A, B and $\xi = \sqrt{A^2 + B^2}$ found as solutions for transform approximation problem.

Factors			Approximation errors		Complexity of multiplications by factors A and B
A	B	$\xi = \sqrt{A^2 + B^2}$	$\cos\left(\frac{3\pi}{8}\right) - \frac{A}{\xi}$	$\sin\left(\frac{3\pi}{8}\right) - \frac{B}{\xi}$	
1	2	$\sqrt{5}$	-0.0645302	0.0294523	1 shift
2	5	$\sqrt{29}$	0.0112928	-0.00459716	1 addition + 2 shifts
3	7	$\sqrt{58}$	-0.0112359	0.0047345	2 additions + 1 shift
5	12	13	-0.00193195	0.000802609	2 additions + 2 shifts
17	41	$\sqrt{1970}$	-0.00033159	0.000137419	3 additions + 2 shifts

Table 2. Complete set of factors selected for the design of 16-point transform.

A	B	C	D	E	F	N	L	J	H	G	I	K	M
2	5	19	4	16	11	6	11	21	27	34	38	42	43

Table 3. Normalized set of factors selected for the design of 16-point transform.

A	B	C	D	E	F	N	L	J	H	G	I	K	M
2/4	5/4	19/32	4/32	16/32	11/32	6/64	11/64	21/64	27/64	34/64	38/64	42/64	43/64

Based on Table 1, it can be observed that larger values A and B produce better approximations, but this also increases complexity (in terms of bit-width and gates count) of the transform. But we also note that even small numbers (such A=2 and B=5) seem to achieve pretty good precision (around 1%) in this case.

We repeat similar exhaustive search for other groups of factors as well. A full set of factors that we have selected for N=16 point transform design is shown in Table 2.

In order to reduce the dynamic range of factors within the transform we further normalize them by nearest dyadic numbers. This optimization is prompted by the fact that divisions by dyadic numbers can be implemented as binary shifts, which are gratis on many platforms. The resulting normalized quantities are now shown in Table 3.

Finally, we now need to compute exact values of scale factors moved outside the transform. In the original form, as shown in Figure 1, such factors are:

$$S = \text{diag} \left(\frac{1}{4}, \frac{1}{2\sqrt{2}\eta}, \frac{1}{4\zeta}, \frac{1}{2\sqrt{2}\xi\eta}, \frac{1}{2\sqrt{2}\xi}, \frac{1}{2\sqrt{2}\xi\eta}, \frac{1}{2\sqrt{2}\zeta}, \frac{1}{4\eta}, \frac{1}{4}, \frac{1}{4\eta}, \frac{1}{2\sqrt{2}\zeta}, \frac{1}{2\sqrt{2}\xi\eta}, \frac{1}{2\sqrt{2}\xi}, \frac{1}{2\sqrt{2}\xi\eta}, \frac{1}{4\zeta}, \frac{1}{2\sqrt{2}\eta} \right);$$

By incorporating scale factor values from our integer solutions (Tables 2), and their subsequent normalization, we obtain:

$$\left[\frac{1}{4}, \frac{10\sqrt{3770}}{1885}, \frac{8}{\sqrt{377}}, \frac{04\sqrt{130}}{1885}, \frac{\sqrt{38}}{29}, \frac{04\sqrt{130}}{1885}, \frac{8\sqrt{734}}{377}, \frac{10}{\sqrt{1885}}, \frac{1}{4}, \frac{10}{\sqrt{1885}}, \frac{8\sqrt{734}}{377}, \frac{04\sqrt{130}}{1885}, \frac{\sqrt{38}}{29}, \frac{04\sqrt{130}}{1885}, \frac{8}{\sqrt{377}}, \frac{10\sqrt{3770}}{1885} \right]$$

After multiplication by 4 and conversion to floating point representation, these scale factors are:

$$[1.0, 2.084679494, 1.648088848, 1.348461033, 1.030431463, 1.348461033, 2.330742331, 1.474091007, 1.0, 1.474091007, 2.330742331, 1.348461033, 1.030431463, 1.348461033, 1.648088848, 2.084679494]$$

which is reasonably tight, considering 4 stages of the transform.

In our final design these factors are cross-multiplied to produce a scale matrix for the entire 2D transform, which is subsequently merged with the quantization matrix of video encoder.

4. EXPERIMENTAL RESULTS

The proposed scheme is tested in the JMKA video encoder under the VCEG common testing conditions²⁵. The reference transforms is a much more complex, direct approximation of matrices of DCT-II transforms by using 8-bit fixed-point factors. Full matrix multiplication is used. The configuration parameters UseRDO_Q, UseHPFilter and UseNewOffset are set to 1 in both the reference and our proposed solution. Extended block sizes²³, MDDT²⁴ (for intra coded macroblocks), and CABAC are also enabled in both the reference as well as our proposed solution. Table 4 shows the percentages of BD-Rate reduction for all test sequences (up to 1080P resolution) under the VCEG common testing conditions. On average, the use of proposed transforms for extended block sizes, results in a 0.0159% and 0.0359% BD-Rate increase for IPPP and HierB configurations.

Table 4: Simulation results for proposed transforms for extended block sizes.

Sequence	Conditions: BigBlocks, MDDT, CABAC, UseRDO_Q=1, UseNewOffset = 1, UseHPFilter = 1,	
	IPPP	HierB
CIF		
Foreman	-0.04	-0.06
Mobile	+0.01	-0.02
Paris	+0.06	+0.02
Tempete	-0.02	+0.08
Average	0.0025	+0.005
WQVGA		
Flower4	-0.20	-0.23
Keiba3	+0.20	+0.05
Nuts5	+0.37	+0.14
Average	+0.1233	-0.0133
WVGA		
Flower4	-0.02	+0.10
Keiba3	-0.38	-0.20
Nuts5	+0.17	+0.29
Average	-0.0767	-0.0633
SVGA		
Janine	+0.15	+0.11
Average	+0.15	+0.11
720P		
BigShips	-0.06	+0.08
City	+0.15	+0.06
Crew	+0.27	+0.13
Night	+0.03	-0.01
Jets	-1.04	-0.02
Raven	+0.16	+0.38
Average	-0.0817	+0.1033
1080P		
CrowdRun	+0.06	-0.03
ParkJoy	+0.06	+0.07
Traffic	-0.15	+0.05
Toysandcalendar	+0.09	+0.16
Subnflower	+0.48	-0.36
Average	0.108	-0.022
Overall Average	+0.0159	+0.0359

5. CONCLUSIONS

We have proposed design of fast orthogonal integer transforms for video coding. The design is based on a simple recursive factorization structure, allowing very compact and efficient implementations. We have implemented and tested our transforms in VCEG's H.265/JMkTA framework. We have showed that under common VCEG test conditions our proposed transforms achieve nearly identical performance compared to much more complex transforms in the current test model.

REFERENCES

- [1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Computers*, X, 90-93 (1974).
- [2] K. R. Rao and P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, Applications," Academic Press (1990).
- [3] V. Britanak, P. Yip, K. R. Rao, "Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations," Academic Press (2006).
- [4] G. J. Sullivan, "Standardization of IDCT approximation behavior for video compression: the history and the new MPEG-C parts 1 and 2 standards," *Proc. SPIE* 6696, pp. 669611 1--22 (2007).
- [5] Y. A. Reznik, A. T. Hinds, C. Zhang, L. Yu, and Z. Ni, "Efficient fixed-point approximations of the 8x8 inverse discrete cosine transform," *Proc. SPIE* 6696, pp. 669617 1--17 (2007).
- [6] ISO/IEC 10918-1 | ITU-T Rec. T.81, "Information Technology – Digital Compression and Coding of Continuous-Tone Still Images – Requirements and Guidelines" (1992).
- [7] ISO/IEC 13818-2 | ITU-T Rec. H.262, "Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video" (1997)
- [8] ITU-T Rec. H.263, "Video Coding for Low Bit Rate Communication" (2005).
- [9] ISO/IEC 14496-2, "Information Technology – Coding of Audio-Visual Objects – Part 2: Visual" (2001).
- [10] ISO/IEC 14496-10 | ITU-T Rec. H.264, "Advanced Video Coding for Generic Audiovisual Services" (2007).
- [11] SMPTE Standard 421M-2006, "VC-1 Compressed Video Bitstream Format and Decoding Process" (2006).
- [12] PRC National Standard (AVS Working Group) GB/T 20090.2-2006, "Information Technology – Advanced Coding of Audio and Video, Part 2: Video" (2006).
- [13] S. Srinivasan, C. Tu, S. L. Regunathan, G. J. Sullivan, and R. A. Rossi, "HD Photo: A New Image Coding Technology for Digital Photography," *Proc. SPIE*, Vol. 6696 (2007).
- [14] ISO/IEC JTC1 SC29 WG11 | ITU-T SG16 Q6 (JCT-VC), "Test Model under Consideration (TMuC)," JCT-VC document JCTVC-A205 (2010).
- [15] W. Chen, C. H. Smith, and S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. Comm.*, 25 (9) 1004-1009 (1977).
- [16] C. Loeffler, A. Ligtenberg, and G. S. Moschytz. "Algorithm-architecture mapping for custom DCT chips," *Proc. International Symposium on Circuits and Systems*, 1953-1956 (1988).
- [17] G. Plonka and M. Tashe, "Fast and numerically stable algorithms for discrete cosine transforms", *Linear Algebra and Applications*, 394 (1) 309-345 (2005).
- [18] Y. A. Reznik, A. T. Hinds, and J. L. Mitchell, "Improved precision of fixed-point algorithms by means of common factors," *Proc. International Conference on Image Processing (ICIP)*, 2344-2347 (2008).
- [19] R. K. Chivukula and Y. A. Reznik, "Efficient implementation of a class of MDCT/IMDCT filterbanks for speech and audio coding applications," *Proc. Int. Conf. Acoustics Speech and Signal Processing*, 213-216 (2008).
- [20] Y. A. Reznik and R. K. Chivukula, "Design of Fast Transforms for High-Resolution and Video Coding," *Proc. SPIE* 7443, 744312 1-17 (2009).
- [21] R. Joshi, Y. Reznik, and M. Karczewicz, "Simplified Transforms for Extended Block Sizes," Input contribution to ITU-T SG16 Q6 (VCEG), document VCEG-AL30 (2009).
- [22] M. Karczewicz, P. Chen, R. Joshi, X. Wang, W.-J. Chien, R. Panchal, Video coding technology proposal by Qualcomm, Input contribution to JCT-VC, document JCTVC-A121 (2010).
- [23] P. Chen, Y. Ye, and M. Karczewicz, "Video coding using extended block sizes," input contribution to ITU-T SG16 Q6 (VCEG), document COM 16-C123-E (2009).
- [24] Y. Ye and M. Karczewicz, "Improved intra coding," input contribution to ITU-T SG16 Q6 (VCEG), document VCEG-AG11 (2007).
- [25] TK Tan, G. Sullivan, and T. Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 4," ITU-T SG16 Q6 (VCEG) document VCEG-AJ10r1 (2008).