Parametric Integer Cosine Transform

Weijia Cao, Yicong Zhou*,
Department of Computer and information science,
University of Macau, Macau, China,
Email: yicongzhou@umac.mo.

***Abstract*—Recent research has shown that integer cosine transforms (ICTs) can avoid some mismatches of discrete cosine transform (DCT) and improve the efficiency of image compression. Furthermore, ICTs with different sizes can be implemented in numerous fields. For example, the larger transforms can improve the transform coding performance in high-definition videos, such as the size of 16×16. This paper generates a technique of parametric integer cosine transform (PICT), which owns the flexibilities of choosing separate integer parameters and different sizes including 16-order. This strategy also has a simple structure based on the matrix factorization, which only contains the integer operation. The proposed ICT can lead to a higher coding gain and peak signal-to-noise ratio performance as well as closer to the original discrete cosine transform than state of the art.**

***Index Terms*—Discrete cosine transform, integer cosine transform, image compression**

I. Introduction

With the explosive increase of using networks and communication technologies, a significant number of images and videos are widely transmitted and stored in the applications. Raw formats of these multimedia always contain a considerable amount of bits, which have much redundant information and lower the efficiency of communicating or storing data. Therefore, in order to shorten the transmission time or the cost of storage, many approaches are built up to compress the digital multimedia, such as predictive coding [1], wavelet coding [2], and transform coding [3].

Among these methods, the block-based transform coding technique shows its extraordinary properties in energy compaction, such as its simplicity, strong compression, and high decorrelation properties. For example, Karhunen Love Transform (KLT) is a linear transform to compress digital images with significantly removing the correlation of neighbouring data [4]. However, it has signal-dependent property and highly computation cost. A much superior alternative to image compression is exercised by the discrete cosine transform (DCT) which is similar to the KLT and widely applied in many coding standards like Moving Picture Experts Group (MPEG) or Joint Photographic Experts Group (JPEG). The DCT is an efficient image compression method which is extruded for its high decorrelation and excellent energy compaction properties in numerous fields such as image compression, filtering and encryption. It can map the image into a set of float coefficients and then codes the quantized coefficients. However, it can also cause some undesirable data loss due to their float coefficients, which are not suitable for the high quality of requiring applications.

To avoid inverse transform discrepancy of the DCT [5], an interesting method, called integer cosine transform (ICT), is proposed by W. K. Cham [6] and recently developed by many researchers. Both of forward and inverse transformations of ICT are comprised of integers, and its kernel is approximated to that of the DCT. Therefore, ICT can save unnecessary float data loss due to its integer property in image compression [7]. Furthermore, its DCT-like, low latency and efficient computational properties are illustrated in [8]. Due to these strengths, many ICTs with different orders are developed for various purposes [7], [9], [10]. For instance, 4-order or 8-order ICT is adopted well in the latest video coding standard H.264/AVC [9], and 16-order ICT can profoundly enhance the coding performance for high-definition (HD) video [11].

In this paper, we develop a parametric ICT (PICT) with arbitrary orders for unique needs. The PICT is an orthogonal ICT which can be recursively represented with integer parameters. It possesses a higher transform coding and peak signal-to-noise ratio (PSNR) performance than the state-of-the-art algorithm during the compressing process. The rest of this paper is organised as follows: Section II reviews the basic ideas of DCT and its scaled form (scaled DCT), and the proposed parametric integer cosine transform (PICT) is presented in Section III. Evaluating the performance of PICT is shown in Section IV. Section V reaches a conclusion.

II. DCT

In this section, we will first review the definitions of DCT and its scaled form. Then it will illustrate a recursively factorization of scaled DCT.

An N-length given input sequence $x_0, x_1, \ldots, x_N$ can be converted into $X_0, X_1, \ldots, X_{N-1}$ by the following equation

$$X(i) = \gamma_i \sum_{j=0}^{N-1} x_j \cos\left(\frac{(2j+1)i\pi}{2N}\right), 0 \le j \le N-1 \quad (1)$$

where

$$\gamma_i = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } i = 0 \\ \sqrt{\frac{2}{N}}, & \text{otherwise} \end{cases} \quad (2)$$

Let $x_N = (x_0, x_1, \ldots, x_N)^T$ and $X_N = (X_0, X_1, \ldots, X_{N-1})^T$ be the input and output signal sequence, respectively. The definition of DCT given by Eq. (1) can be represented in a matrix form as

$$X_N = C_N x_N, \quad (3)$$

978-1-5386-2201-8/17/$31.00 ©2017 IEEE

where C_N is the matrix of DCT expressed as

$$C_N(i,j) = \gamma_i \cos\left[\frac{(2j+1)i\pi}{2N}\right], 0 \le i,j \le N-1 \quad (4)$$

where γ_i is defined in Eq. (2).

In order to simplify the computation process and reduce the complexity, the researchers always use the scaled DCT which removed the irrational parameter (i.e. the scaled factor γ_i). Then the matrix of scaled DCT can be represented as

$$S_N = \cos\left[\frac{(2j+1)i\pi}{2N}\right]. \quad (5)$$

It has been demonstrated that the scaled DCT matrix S_{2N} can be factorized into odd part S_N and even part F_N as [12]

$$S_{2N} = P_{2N}\begin{bmatrix} S_N & 0 \\ 0 & F_N J_N \end{bmatrix}\begin{bmatrix} I_N & J_N \\ J_N & -I_N \end{bmatrix} \quad (6)$$

where P_{2N} is a permutation matrix, I_N is an N-order identity matrix, and J_N is an N-order identity matrix by reversing the rows of I_N. For any input vector $V_{2N} = (V_0, V_1, ..., V_{2N-1})^T$, the output $P_{2N}V_{2N} = (V_0, V_N, V_1, V_{N+1}, ..., , V_{N-1}, V_{2N-1})^T$.

III. Parametric Integer Cosine Transform

In this section, we propose a block-based parametric integer cosine transform (PICT), which utilizes the factorization of DCT in Eq. (6). The kernel R_{2N} of a $2N$-order PICT can be represented as

$$R_{2N} = P_{2N}\begin{bmatrix} R_N & 0 \\ 0 & G_N J_N \end{bmatrix}\begin{bmatrix} I_N & J_N \\ J_N & -I_N \end{bmatrix} \quad (7)$$

where odd part R_N is a recursive matrix with integer parameters, the even part G_N is an integer symmetric matrix, and the definitions of other matrices are same as that in Eq. (6). Here we can set R_2 and G_2 as

$$R_2 = \begin{bmatrix} a_1 & a_1 \\ a_2 & -a_2 \end{bmatrix}, G_2 = \begin{bmatrix} a_3 & a_4 \\ a_4 & -a_3 \end{bmatrix}, \quad (8)$$

where a_1, a_2, a_3, and a_4 are positive integers. The default values are set as $a_1 = 256$, $a_2 = 229$, $a_3 = 320$, and $a_4 = 128$.

Therefore, in Eq. (7), R_{2N} can be recursively represented if the even part G_N is determined. Then it is necessary to decide the function of the matrix G_N. However, G_N is difficult to be factorized and then it can be decomposed into small element under some constraints for simplicity. The basic structure of it can be represented as

$$G_N = \begin{bmatrix} A_{N/2} & \bar{J}_{N/2}B_{N/2}\bar{J}_{N/2} \\ B_{N/2} & \bar{J}_{N/2}A_{N/2}\bar{J}_{N/2} \end{bmatrix}, \quad (9)$$

where A and B are parametric integer matrices. $\bar{J}_{N/2}$ is defined as

$$\bar{J}_{N/2} = \begin{bmatrix} & & & & 1 \\ & & & -1 & \\ & & \iddots & & \\ & 1 & & & \\ -1 & & & & \end{bmatrix}. \quad (10)$$

When $N = 4$, the matrices A_2 and B_2 can be written as

$$A_2 = \begin{bmatrix} a & b \\ b & -d \end{bmatrix}, B_2 = \begin{bmatrix} c & -a \\ d & -c \end{bmatrix}, \quad (11)$$

where a, b, c, d are non-negative integers with the constraint $ab - bd - ac - cd = 0$. When $a = 5, b = 3, c = 2, d = 1$, G_4 can be represented as

$$G_4 = \begin{bmatrix} 5 & 3 & 2 & 1 \\ 3 & -1 & -5 & -2 \\ 2 & -5 & 1 & 3 \\ 1 & -2 & 3 & -5 \end{bmatrix}.$$

Then the matrices A_4 and B_4 can be written as

$$A_4 = \begin{bmatrix} e & f & g & h \\ f & i & j & -l \\ g & j & -h & -f \\ h & -l & -f & j \end{bmatrix}, B_4 = \begin{bmatrix} i & -g & -m & e \\ l & -e & i & m \\ m & -h & e & -g \\ j & -m & l & -i \end{bmatrix},$$

where e, f, g, h, i, j, l, m are non-negative integers with the constraint to make G_8 as an orthogonal matrix.

For example, when $e = 228, f = 222, g = 213, h = 188, i = 148, l = 111, m = 60, j = 36$, G_8 can be represented as

$$G_8 = \begin{bmatrix} 228 & 222 & 213 & 188 & 148 & 111 & 60 & 36 \\ 222 & 148 & 36 & -111 & -213 & -228 & -188 & -60 \\ 213 & 36 & -188 & -222 & -60 & 148 & 228 & 111 \\ 188 & -111 & -222 & 36 & 228 & 60 & -213 & -148 \\ 148 & -213 & -60 & 228 & -36 & -222 & 111 & 188 \\ 111 & -228 & 148 & 60 & -222 & 188 & 36 & -213 \\ 60 & -188 & 228 & -213 & 111 & 36 & -148 & 222 \\ 36 & -60 & 111 & -148 & 188 & -213 & 222 & -228 \end{bmatrix}.$$

Even if we choose a few of different parameters, the matrix of PICT can be various. For simplicity, we select three examples with just a few different parameters a, b, c, d of A_2 and B_2 in Eq. (11). Then we name them as (PICT-R1)$_N$, (PICT-R2)$_N$, and (PICT-R3)$_N$ to represent N-order PICT matrix in Table I , respectively.

TABLE I
Examples of N-order PICTs.

N-order PICTs	Parameters			
	a	b	c	d
(PICT-R1)$_N$	430	369	246	86
(PICT-R2)$_N$	465	399	266	93
(PICT-R3)$_N$	378	325	215	77

Then the PICT-R1$_{16}$ can be represented in Eq. (12) after the recursive factorization in Eq. (7).

In order to maintain the orthogonality of transform, the kernel R_{2N} needs a normalization for their basis vectors. Then the normalized kernel W_{2N} of PICT can be obtained from R_{2N} as

$$W_{2N} = L_{2N}R_{2N}, \quad (13)$$

where L_{2N} is a diagonal matrix with certain parameters which can convert the basis vectors of corresponding the matrices of PICT into unit ones. As is a unit orthogonal matrix, the inverse of W_{2N} is its transposition. In the image compressing process, L_{2N} can be put in the quantization step to enhance the efficiency of compressing computation.

$$\text{PICT-R1}_{16} = \begin{bmatrix} 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 & 256 \\ 228 & 222 & 213 & 188 & 148 & 111 & 60 & 36 & -36 & -60 & -111 & -148 & -188 & -213 & -222 & -228 \\ 430 & 369 & 246 & 86 & -86 & -246 & -369 & -430 & -430 & -369 & -246 & -86 & 86 & 246 & 369 & 430 \\ 222 & 148 & 36 & -111 & -213 & -228 & -188 & -60 & 60 & 188 & 228 & 213 & 111 & -36 & -148 & -222 \\ 320 & 128 & -128 & -320 & -320 & -128 & 128 & 320 & 320 & 128 & -128 & -320 & -320 & -128 & 128 & 320 \\ 213 & 36 & -188 & -222 & -60 & 148 & 228 & 111 & -111 & -228 & -148 & 60 & 222 & 188 & -36 & -213 \\ 369 & -86 & -430 & -246 & 246 & 430 & 86 & -369 & -369 & 86 & 430 & 246 & -246 & -430 & -86 & 369 \\ 188 & -111 & -222 & 36 & 228 & 60 & -213 & -148 & 148 & 213 & -60 & -228 & -36 & 222 & 111 & -188 \\ 229 & -229 & -229 & 229 & 229 & -229 & -229 & 229 & 229 & -229 & -229 & 229 & 229 & -229 & -229 & 229 \\ 148 & -213 & -60 & 228 & -36 & -222 & 111 & 188 & -188 & -111 & 222 & 36 & -228 & 60 & 213 & -148 \\ 246 & -430 & 86 & 369 & -369 & -86 & 430 & -246 & -246 & 430 & -86 & -369 & 369 & 86 & -430 & 246 \\ 111 & -228 & 148 & 60 & -222 & 188 & 36 & -213 & 213 & -36 & -188 & 222 & -60 & -148 & 228 & -111 \\ 128 & -320 & 320 & -128 & -128 & 320 & -320 & 128 & 128 & -320 & 320 & -128 & -128 & 320 & -320 & 128 \\ 60 & -188 & 228 & -213 & 111 & 36 & -148 & 222 & -222 & 148 & -36 & -111 & 213 & -228 & 188 & -60 \\ 86 & -246 & 369 & -430 & 430 & -369 & 246 & -86 & -86 & 246 & -369 & 430 & -430 & 369 & -246 & 86 \\ 36 & -60 & 111 & -148 & 188 & -213 & 222 & -228 & 228 & -222 & 213 & -188 & 148 & -111 & 60 & -36 \end{bmatrix} \quad (12)$$

IV. Transform Evaluation

This section will assess the performance of PICT by some comparisons regarding transform coding gain, mean square error, and peak signal-to-noise ratio. It will be compared with traditional DCT and state-of-art integer cosine transforms, such as ICT [13], and LLM [10].

A. Transform coding gain

To estimate the coding efficiency of an integer transform, transform coding gain plays a significant part in the quantized evaluation.

$$M_{cg} = 10\log_{10}\frac{1}{N}\sum_{i=0}^{N-1}\alpha_i^2/(\prod_{i=0}^{N-1}\alpha_i^2\|b_i\|^2)^{\frac{1}{N}}, \qquad (14)$$

where N is the order of the transform, α_i^2 is the variance of the transform coefficient, and $\|b_i\|^2$ is the 2-norm of i^{th} basis function of the transform matrix. The typical domain of α is $[0.5, 0.9]$. Transform can compress more information into fewer number of parameters with a higher W_{cg}. Table II displays the transform coding gains between traditional DCT, ICT [13], LLM [10], and the examples of PICT in Table II.

TABLE II
Transform coding gains of different integer cosine transforms with the size of 16×16.

Transform Types	Transform coding gain $M_{cg}(dB)$				
	ρ = 0.5	ρ = 0.6	ρ = 0.7	ρ = 0.8	ρ = 0.9
DCT	1.1414	1.7794	2.6982	4.1150	6.7263
ICT [13]	1.1348	1.7690	2.6816	4.0888	6.6853
LLM [10]	1.1405	1.7778	2.6947	4.1081	6.7124
$(\text{PICT-R1})_{16}$	**1.1405**	**1.7779**	**2.6953**	**4.1090**	**6.7126**
$(\text{PICT-R2})_{16}$	**1.1405**	**1.7779**	**2.6953**	**4.1090**	**6.7126**
$(\text{PICT-R3})_{16}$	1.1405	1.7778	2.6947	4.1081	6.7124

As can be seen in Table II, the coding gain results of PIDCT are much higher and closer to that of traditional DCTs than other 16-order transforms.

B. Distortion Comparison

The ICTs are always designed to be approximate to the traditional DCTs. Then mean square errors can indicate the difference between the traditional DCT and the proposed one. The smaller one is always a better performance. Here we take DCT, LLM [10], and PICT with 8-order in Table III, respectively.

TABLE III
Mean square errors of different integer cosine transforms with the size of 8×8.

Transform Types	Mean square errors
DCT	0
LLM [10]	$4.155 * 10^{-4}$
$(\text{PICT-R1})_8$	$6.747 * 10^{-6}$
$(\text{PICT-R2})_8$	**$6.703*10^{-6}$**
$(\text{PICT-R3})_8$	$6.0521 * 10^{-5}$

We can see from the Table III, $(\text{PICT-R1})_8$ is the closest one of these matrices in the traditional DCT with the smallest MSE.

C. Comparison of Peak Signal-to-Noise Ratios

In practice, images often become blurred or unclear to a certain degree after the lossy compression process in which the quantization step can cut down some redundant information as well as a few useful ones. The peak signal-to-noise ratio (PSNR) is a significant method to measure the approximation between the compressed image and the original image [14]. Then we take a 256×256 grayscale image $Lena$ for an example.

The results compressed by JPEG are shown in Fig. 1. The PSNRs of both LLM [10] and PICT with different quantization factors. With the higher quantization factor, the decline of PSNRs in PICT is slower than that of LLM, which means that the images compressed by PICT are clearer than LLM. In Fig. 2, the PSNR of PICT is higher than that of LLM with the same quantization factor of four.

V. Conclusion

This paper proposed a block-based parametric integer cosine transform (PICT), which can be recursively represented with distinct integer parameters and various sizes. The factorization of PICT can be easily carried out in hardware since there are only integer operations needed. The analysis also illustrated that PICT possesses a higher coding gain and PSNR performance with more approximated to the traditional DCT than state of the art.

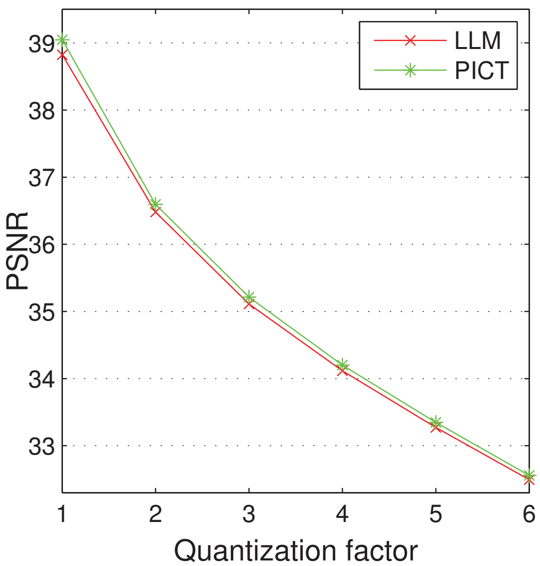

Fig. 1. PSNR comparison between LLM [10] and PICT with different quantization factors. The red and green lines represent LLM and PICT, respectively.

(a) (b) (c)

Fig. 2. Original grayscale image $lena$ with 256×256 (a) compressed by JPEG (Quantization factor is 4) using the integer cosine transforms: (b) LLM (PSNR is 34.1153) and (c) $(\text{PICT-R1})_8$ (PSNR is 34.2006).

ACKNOWLEDGMENT

This work was supported in part by the Macau Science and Technology Development Fund under Grant FDC-T/016/2015/A1 and by the Research Committee at University of Macau under Grants MYRG 2014-00003-FST and MYRG 2016-00123-FST.

REFERENCES

[1] J. A. Robinson, "Efficient general-purpose image compression with binary tree predictive coding," *IEEE Transactions on Image Processing*, vol. 6, no. 4, pp. 601–608, Apr 1997.

[2] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, Jun 1996.

[3] C. K. Fong, Q. Han, and W. K. Cham, "Recursive integer cosine transform for hevc and future video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 2, pp. 326–336, Feb 2017.

[4] A. Jain, "A fast karhunen-loeve transform for a class of random processes," *IEEE Transactions on Communications*, vol. 24, no. 9, pp. 1023–1029, Sep 1976.

[5] C. Zhang, L. Yu, J. Lou, W. K. Cham, and J. Dong, "The technique of prescaled integer transform: Concept, design and applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 84–97, Jan 2008.

[6] W. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," in *Communications, Speech and Vision, IEE Proceedings I*, vol. 136, no. 4. IET, 1989, pp. 276–282.

[7] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[8] G. A. Ruiz, J. A. Michell, A. M. Buron, J. M. Solana, M. A. Manzano, and J. Díaz, "Integer cosine transform chip design for image compression," in *Microtechnologies for the New Millennium 2003*. International Society for Optics and Photonics, 2003, pp. 33–41.

[9] D. Marpe, T. Wiegand, and G. J. Sullivan, "The h.264/mpeg4 advanced video coding standard and its applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–143, Aug 2006.

[10] C. K. Fong and W. K. Cham, "Llm integer cosine transform and its fast algorithm," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 6, pp. 844–854, June 2012.

[11] S. Ma and C.-C. J. Kuo, "High-definition video coding with super-macroblocks." International Society for Optics and Photonics, 2007.

[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 and Y. T. Chan, "An order-16 integer cosine transform," *IEEE Transactions on Signal Processing*, vol. 39, no. 5, pp. 1205–1208, May 1991.

[14] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards-Including High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, Dec 2012.