

# 2-D Order-16 Integer Transforms for HD Video Coding

Jie Dong, *Student Member, IEEE*, King Ngai Ngan, *Fellow, IEEE*, Chi-Keung Fong, *Student Member, IEEE*,  
and Wai-Kuen Cham, *Senior Member, IEEE*

**Abstract**—In this paper, the spatial properties of high-definition (HD) videos are investigated based on a large set of HD video sequences. Compared with lower resolution videos, the prediction errors of HD videos have higher correlation. Hence, we propose using 2-D order-16 transforms for HD video coding, which are expected to be more efficient to exploit this spatial property, and specifically propose two types of 2-D order-16 integer transforms, nonorthogonal integer cosine transform (ICT) and modified ICT. The former resembles the discrete cosine transform (DCT) and is approximately orthogonal, of which the transform error introduced by the nonorthogonality is proven to be negligible. The latter modifies the structure of the DCT matrix and is inherently orthogonal, no matter what the values of the matrix elements are. Both types allow selecting matrix elements more freely by releasing the orthogonality constraint and can provide comparable performance with that of the DCT. Each type is integrated into the audio and video coding standard (AVS) Enhanced Profile (EP) and the H.264 High Profile (HP), respectively, and used adaptively as an alternative to the 2-D order-8 transform according to local activities. At the same time, many efforts have been devoted to further reducing the complexity of the 2-D order-16 transforms and specially for the modified ICT, a fast algorithm is developed and extended to a universal approach. Experimental results show that 2-D order-16 transforms provide significant performance improvement for both AVS Enhanced Profile and H.264 High Profile, which means they can be efficient coding tools especially for HD video coding.

**Index Terms**—AVS, H.264, HDTV, ICT, order-16 transform, VBT.

## I. INTRODUCTION

TODAY, high-definition (HD) videos become more and more popular with many applications, such as high-definition television (HDTV) broadcasting, high density storage media, video-on-demand (VOD), and surveillance. The most common algorithms for HD video coding are mainly based on the hybrid framework of motion compensated prediction (MCP) and discrete cosine transform (DCT).

Manuscript received July 13, 2007; revised April 28, 2008. First version published July 7, 2009; current version published September 30, 2009. This work was supported in part by the Chinese University of Hong Kong, Hong Kong, China under the Focused Investment Scheme, Project 1903003, and the Research Grants Council of Hong Kong, Project CUHK410307. This paper was recommended by Associate Editor P. Topiwala.

The authors are with the Department of Electronic Engineering, the Chinese University of Hong Kong, Hong Kong, China (e-mail: jdong@ee.cuhk.edu.hk; knngan@ee.cuhk.edu.hk; ckfong@ee.cuhk.edu.hk; wkcham@ee.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2009.2026792

MPEG-2 [1] is the most widely used standard for HDTV broadcasting, which uses only basic coding tools, such as the 2-D order-8 DCT, bidirectional motion prediction and compensation, and variable length coding (VLC). Although the efficiency is relatively low, the widespread acceptance of MPEG-2 is mainly attributed to its low complexity and its adoption in consumer electronic products. The latest video coding standard H.264/AVC [2] can bring significant performance improvement compared to all the previous standards. It employs complicated intra and inter prediction tools to achieve accurate prediction and advanced entropy coding schemes to remove statistical redundancy. The recently completed High Profile (HP) [3], aiming at HD video coding, further improves the coding efficiency by variable block size transforms (VBT), where a 2-D order-8 integer cosine transform (ICT) is used adaptively as an alternative to the 2-D order-4 ICT in H.264. The audio and video coding standard (AVS) [4] is the national standard of China. Its Enhanced Profile (EP) [5] also aims at HD video coding in the efficiency aspect and does not provide rich functionalities. So, it is quite similar to the Main Profile of H.264. JPEG 2000 [6] is another attractive choice for HD video coding. Compared with those MCP/DCT-based coding techniques, it provides a wavelet-based framework and offers the richest set of features, such as lossy and lossless coding, spatial and quality scalability, region-of-interest (ROI) coding, and error resilience tools. Although it was originally developed for still image coding and does not remove temporal redundancy of video sequences, JPEG 2000 can address the needs for very demanding HD applications, e.g., digital cinema, and remote medical consultancy.

However, the techniques, focusing on improving coding efficiency, are all originally designed as general coding tools for videos of various resolutions. Although efficient when applied to HD videos, they do not fully exploit their properties. Wien and Sun [7] once proposed a VBT scheme to H.264, with seven transform block sizes of  $4 \times 4$  up to  $16 \times 16$ . However, this scheme was not proposed for HD video coding and the reported experimental results were based on a set of CIF sequences. The maximum transform block size was finally reduced to  $8 \times 8$  due to the visible artifacts and the overall complexity [8]. Recently, Naito and Koike [9], and Ma and Kuo [10] have investigated the efficiency of using super-macroblock (MB) for HD video coding within the framework of H.264. Naito and Koike [9] allow the MB size to be adaptively selected from  $16 \times 16$  to  $2^N \times 2^N$  ( $N \geq 4$ ), where  $N$

may be configured at the picture level. The block sizes of intra prediction and motion compensation are scaled, according to the MB size. The experimental results based on two  $4K \times 2K$  sequences are reported and significant bit saving can be observed, when the sequences are coded using the fixed QP 45. In [10], the MB size is fixed to be  $32 \times 32$  and a 2-D order-16 integer transform is also proposed and adaptively used in addition to the 2-D order-4 and order-8 ICTs in H.264 HP. Modest PSNR gain is shown based on two 1080p sequences, but subjective improvement is invisible.

In this paper, we study the unique properties of HD videos and thus develop special coding tools. First of all, the spatial properties of HD videos and their motion prediction errors are both investigated and compared with those of low resolution videos. It is found that the HD videos are distinguished from other low resolution ones by higher spatial correlation. To exploit the property, we propose using 2-D order-16 transforms for HD video coding. Specifically, two types of 2-D order-16 integer transforms are proposed, both of which can achieve comparable performance with the DCT, if well-designed, and provide the freedom of selecting matrix elements by releasing the orthogonality constraint, such that trade-off can be made between performance and complexity.

One type, called nonorthogonal ICT (NICT), allows the basis vectors to be nonorthogonal, in order to use matrix elements with small magnitudes on one hand and maintain the structure of the DCT matrix, e.g., dyadic symmetries and relative magnitudes, on the other hand. The transform error, which makes the average variance of the reconstruction error larger than that of the quantization error due to the nonorthogonality, is analyzed quantitatively. The specific NICT proposed in this paper achieves the balance among the performance, the complexity, and the transform error.

The other type, called modified ICT (MICT), is an orthogonal integer transform with the dyadic symmetries of the DCT matrix modified. It is naturally orthogonal, no matter what the values of the elements are. There are some related publications. The one proposed by Wien and Sun [7] has a one-norm transform matrix, which means the norms of all the basis vectors are the same and the memory for storing the scaling matrix can be saved. However, as the cost, this 2-D order-16 integer transform has much lower coding gain [11] than the DCT and introduces ringing artifacts, when integrated into H.264. The one proposed by Ma and Kuo [10] is simple and compatible with the 2-D order-8 ICT in H.264. However, the proposed transform modifies the structure of the DCT matrix so significantly that it is actually equivalent to the combined processes of 2-D order-8 ICTs and 2-D order-2 Hadamard Transforms (HT). In other words, first, the four  $8 \times 8$  partitions of the input  $16 \times 16$  block are transformed by 2-D order-8 ICT, respectively, and then the coefficients at the corresponding positions of the four transformed blocks are transformed by 2-D order-2 HT and thus further de-correlated. Liang *et al.* [12], [13] proposed a family of approximations of the DCT, named binDCT, and the lifting scheme for implementation. The lifting parameters, which are theoretically rational, are replaced by proper dyadic approximations and thus enable the multiplication-free fast algorithm. The trade-off between

coding gain and computational complexity can be made by tuning the resolution of the approximation. However, the binDCT is not a unitary transform and thus uses the exact inversion, although invertibility is guaranteed by the lifting scheme. As a drawback, matrix multiplication cannot provide the exact same results as the lifting scheme, unless additional shifts are used.

The proposed two types of 2-D order-16 transforms are both integrated into AVS EP and H.264 HP, respectively, using 16-bit integer arithmetic and used adaptively as an alternative to the 2-D order-8 ICT in whichever standard they are integrated into. The related problems of transform size selection,  $16 \times 16$  intra prediction, entropy coding, and coded block pattern (CBP) are solved. Both 2-D order-16 integer transforms are compatible with the 2-D order-8 ones in the standards. The fast algorithm for the MICT is developed and extended to a general approach.

The remainder of this paper is organized as follows. Section II investigates the unique spatial property of HD videos. To fully exploit the property, two types of 2-D order-16 integer transforms are proposed in Section III. Section IV introduces the efforts we have made to reduce the complexity of the proposed transforms. Section V briefly presents the solutions of the related problems, when the 2-D order-16 transforms are integrated into the standards. Section VI reports the experimental results, followed by the conclusion in Section VII.

## II. SPATIAL ANALYSIS OF HD VIDEOS

The most useful statistical function for all video coding methods is correlation. Spatial correlation is useful for estimating the energy of the coefficients in the case of unitary transform coding, whilst temporal correlation can help to estimate the prediction error and design the predictor for interframe coder [14]. Our research work starts from analyzing the spatial correlation of raw HD videos. Since all the transforms used for video coding are separable and the spatial property of columns and rows can be taken as equivalent [15], we only focus on the horizontal correlation.

In a video frame, each row is regarded as an observation of a discrete stochastic process and the observations in total are the rows in one frame. So each column represents a random variable. In probability theory and statistics, correlation of a stochastic process is often measured by the function, correlation coefficient  $r$ , as defined below [16]

$$r(n_1, n_2) = \mathcal{E}[(x(n_1) - \mu_1) \cdot (x(n_2) - \mu_2)] / (\sigma_1 \cdot \sigma_2) \quad (1)$$

where  $n_1$  and  $n_2$  are two positions in a certain row,  $x(n_1)$  and  $x(n_2)$  are the values of pixels at the two positions with ensemble averages,  $\mu_1$  and  $\mu_2$ , and standard deviations,  $\sigma_1$  and  $\sigma_2$ , and  $\mathcal{E}$  is the ensemble average operator. Due to the variable local activities in a frame,  $r(n, n + \tau)$  depends not only on  $\tau$  but also on  $n$  in different positions. In other words, a video frame is not a collection of samples of a wide-sense stationary (WSS) stochastic process. Hence, for a given  $\tau$ ,  $r(\tau)$  is a random variable instead of a constant.

We calculate  $r(\tau)$  with  $\tau=1$ , which represents the correlation of adjacent pixels. Table I shows the expected value

TABLE I

THE CORRELATION COEFFICIENT OF TWO ADJACENT PIXELS IN RAW VIDEO SEQUENCES

Test Sequence	Resolution	$r(1)$	
		$\mu_{r1}$	$\sigma_{r1}$
<i>Riverbed</i>	HD 1920 × 1080	0.9894	0.0042
<i>Flamingo</i>		0.9882	0.0063
<i>Kayak</i>		0.9864	0.0065
<i>Fireworks</i>	HD 1280 × 720	0.8159	0.0441
<i>Raven</i>		0.9770	0.0129
<i>Night</i>		0.9548	0.0511
<i>Crew</i>		0.9758	0.0460
<i>City</i>		0.9513	0.0456
<i>FlowerGarden</i>		0.9417	0.1154
<i>F1</i>	SD 720 × 576	0.8923	0.1439
<i>Basketball</i>		0.9391	0.1066
<i>Mobile</i>		0.8836	0.1154
<i>Foreman</i>	CIF 352 × 288	0.9722	0.0712
<i>News</i>		0.9199	0.1259
<i>Tempete</i>		0.8815	0.0478
<i>Paris</i>		0.8656	0.1053

and standard deviation of  $r(1)$ . A frame in a video sequence is a still image that can be approximated by the first-order Markov source with correlation coefficient tending to 1.0 [15]. Therefore, as what we expected, for all resolution sequences, the mean values of  $r(1)$  approach the upper bound of 1.0 and the variances are very small. Though, overall,  $r(1)$  of HD videos is slightly higher than others, the difference is not substantial.

However, the transforms in video codecs are usually applied to motion prediction errors instead of original pixels. Even for intra frames, prediction errors are transformed thanks to the intra predictions in the recent video coding standards, such as H.264 and AVS. Therefore, it makes sense to investigate the correlation of prediction errors, which influence coding strategies directly. In this paper, the prediction errors are obtained using the criterion of rate–distortion (R–D) cost [17] and only the horizontal correlation is analyzed, as the correlation equivalence of two spatial directions still holds for prediction errors, according to our experimental results.

Table II shows the expected values,  $\mu_{r1}$  and  $\mu_{r2}$ , of  $r(1)$  and  $r(2)$  of prediction errors. Generally, the correlation of prediction errors increases with the resolution, and the difference of  $r(1)$  values between HD and low resolution videos is substantial. The mean values of  $r(2)$ , the correlation of every other pixel, are relatively high for HD videos, but approach 0 for most of the low resolution videos. Therefore, as evident in Table II, the prediction errors of HD videos are distinguished from the low resolution ones by higher spatial correlation. However, this conclusion does not hold in some extreme cases. For example, if a human face in sequence *Crew* was captured by a CIF video, the spatial correlation of the CIF would be higher than the HD. However, based on a large set of

TABLE II

THE CORRELATION COEFFICIENTS  $r(1)$  AND  $r(2)$  OF PREDICTION ERRORS OF VIDEO SEQUENCES

Test Sequence	Resolution	$r(1)$		$r(2)$	
		$\mu_{r1}$	$\sigma_{r1}$	$\mu_{r2}$	$\sigma_{r2}$
<i>Riverbed</i>	HD 1920 × 1080	0.8909	0.1658	0.7570	0.1891
<i>Flamingo</i>		0.8606	0.1596	0.6835	0.1869
<i>Kayak</i>		0.8596	0.1320	0.6630	0.1267
<i>Tractor</i>		0.7804	0.1504	0.5174	0.1968
<i>Pedestrians</i>		0.7772	0.1862	0.5473	0.2601
<i>Station</i>		0.6814	0.1793	0.3551	0.3200
<i>Fireworks</i>		0.5031	0.1285	0.1005	0.1325
<i>Raven</i>		0.7328	0.1174	0.3746	0.1813
<i>Optis</i>		0.6966	0.0698	0.2916	0.1043
<i>Night</i>		HD 1280 × 720	0.6807	0.1660	0.3196
<i>Crew</i>	0.6712		0.1476	0.2647	0.2329
<i>Harbour</i>	0.5738		0.1859	0.0749	0.2559
<i>City</i>	0.5618		0.1080	0.1015	0.1239
<i>FlowerGarden</i>	SD 720 × 576	0.5774	0.1423	0.2140	0.1504
<i>F1</i>		0.5629	0.1809	0.2304	0.2473
<i>Basketball</i>		0.4912	0.2401	0.1888	0.2461
<i>Mobile SD</i>		0.3729	0.1704	0.0875	0.1871
<i>Foreman</i>	CIF 352 × 288	0.5445	0.2035	0.1725	0.2939
<i>News</i>		0.3543	0.1870	0.0466	0.1960
<i>Tempete</i>		0.3176	0.2353	0.2049	0.2182
<i>Mobile</i>		0.2323	0.1686	−0.0539	0.1617
<i>Paris</i>		0.1770	0.2223	0.0798	0.1588

HD videos, the conclusion is useful and quite instructive for studying the tools specially for coding HD videos.

For a given bit rate, the MSE produced by transform coding improves with the block size, but the improvement is not significant as the block size increases beyond  $16 \times 16$  [18]. In other words, compared with order-8 transforms, order-16 transforms compact energy to a smaller proportion of coefficients, which are more likely to survive the quantization. In practice, for image or video coding, the performance of 2-D order-16 transform suffers nonoptimal entropy coding (typically run-length coding), as it is difficult to accurately model the distribution of a *run*, which has a much larger dynamic range than that produced by order-8 transforms and dramatically varies throughout video sequences. Besides, order-16 transform is much more computationally complex, so order-8 transform is a good trade-off in the current image and video coding standards. However, based on the aforementioned observation that the prediction errors of HD videos have higher correlation, 2-D order-16 transforms deserve more study as a special coding tool, as the transform coefficients have high probabilities to be distributed in the low frequency domain and thus *run* can be modeled more accurately.

Table II also shows the standard deviations,  $\sigma_{r1}$  and  $\sigma_{r2}$ , of  $r(1)$  and  $r(2)$  of prediction errors, where there is no significant difference between HD and low resolution videos. So, the

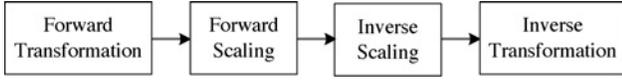


Fig. 1. Flow diagram of traditional ICTs.

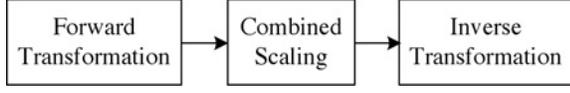


Fig. 2. Flow diagram of the pre-scaled integer transform.

spatial correlation of prediction errors varies greatly within an HD video sequence, just like that in low resolution videos. To exploit such a property, VBT should be employed, in which smaller transforms are the complements to the 2-D order-16 one, in order to reduce the complexity and avoid the ringing artifacts especially in more detailed areas.

### III. THE PROPOSED 2-D ORDER-16 INTEGER TRANSFORMS

#### A. Review of ICT

ICT is generated from the DCT by replacing the real-numbered elements of the DCT matrix with integers while maintaining the structure, such as relative magnitudes and signs, dyadic symmetries, and orthogonality, among the matrix elements [19]. ICT can be implemented using integer arithmetic without mismatch between the encoder and decoder and if well-designed, can provide almost the same compression efficiency as the DCT. Furthermore, ICT has fast algorithms, which can be developed in the similar way for the DCT.

The flow diagram of ICT from the input data to the reconstructed data is shown in Fig. 1. As the transform matrix of ICT contains only integers and the norms of basis vectors are much larger than unity, the transformation process shown as (2) is not normalized. Therefore, a normalization process in (3), known as scaling, is required after transformation to ensure the conservation of energy and the reversibility of ICT

$$F_{n \times n} = T_n \times f_{n \times n} \times T_n^T \quad (2)$$

$$S_{n \times n} = F_{n \times n} // R_{n \times n}. \quad (3)$$

In (2) and (3),  $f_{n \times n}$  is the input data,  $T_n$  is the  $n \times n$  transform matrix,  $S_{n \times n}$  is the output of ICT and symbol // indicates that each element of the left matrix is divided by the

element at the same position of the right one. The elements in the scaling matrix  $R_{n \times n}$  are derived from the norms of basis vectors of  $T_n$  as below, where vector  $m_{n \times 1}$  contains the norms of all basis vectors

$$R_{n \times n} = m_{n \times 1} \times m_{n \times 1}^T \quad (4)$$

$$m_{n \times 1}(i) = \sqrt{\sum_{j=0}^{n-1} T_n^2(i, j)}, \quad 0 \leq i < n. \quad (5)$$

The divisions in (3) are usually approximated by integer multiplication and shift as shown in (6)

$$S_{n \times n} = F_{n \times n} \otimes P_{n \times n} \gg N \quad (6)$$

where symbol  $\otimes$  indicates that each element of the left matrix is multiplied by the element at the same position of the right one and  $\gg N$  means the  $N$ -bit right shift.

Similarly, for the inverse ICT, the whole process including inverse scaling and inverse transformation can be represented by (7) as follows:

$$f_{n \times n} = (T_n^T \times (S_{n \times n} \otimes P_{n \times n}) \times T_n) \gg N. \quad (7)$$

The block diagram shown in Fig. 1 is further simplified by the pre-scaled integer transform (PIT) (see Fig. 2), which moves the inverse scaling from the decoder to the encoder and combines it with the forward scaling as one single process, named combined scaling. With PIT, the inverse scaling is saved and thus no scaling matrix is needed to be stored, whilst the complexity of encoders remains unchanged. However, the inverse scaling moved to the encoder side can be considered as a frequency weighting matrix to the transformed signals. In order not to alter the energy distribution of the transformed signals significantly, the elements' magnitudes in the scaling matrix should be almost the same. In other words, the norms of the basis vectors should be very close to each other according to (4) and (5), which is a key for PIT designing. Interested readers may refer to [20] for more details.

#### B. The Proposed Order-16 NICT

The general transform matrix of an order-16 ICT is shown in (8), at the bottom of the page [19], which has alternating even and odd symmetry with respect to the line. The basis vectors with even symmetry form the even part of the transform matrix,  $T_{8e}$ , whereas those with odd symmetry form the odd

$$T_{16} = \begin{bmatrix} x_0 & \cdots \\ x_1 & x_3 & x_5 & x_7 & x_9 & x_{11} & x_{13} & x_{15} & -x_{15} & -x_{13} & \cdots \\ x_2 & x_6 & x_{10} & x_{14} & -x_{14} & -x_{10} & -x_6 & -x_2 & -x_2 & -x_6 & \cdots \\ x_3 & x_9 & x_{15} & -x_{11} & -x_5 & -x_1 & -x_7 & -x_{13} & x_{13} & x_7 & \cdots \\ x_4 & x_{12} & -x_{12} & -x_4 & -x_4 & -x_{12} & x_{12} & x_4 & x_4 & x_{12} & \cdots \\ x_5 & x_{15} & -x_7 & -x_3 & -x_{13} & x_9 & x_1 & x_{11} & -x_{11} & -x_1 & \cdots \\ x_6 & -x_{14} & -x_2 & -x_{10} & x_{10} & x_2 & x_{14} & -x_6 & -x_6 & x_{14} & \cdots \\ x_7 & -x_{11} & -x_3 & x_{15} & x_1 & x_{13} & -x_5 & -x_9 & x_9 & x_5 & \cdots \\ x_8 & -x_8 & -x_8 & x_8 & x_8 & -x_8 & -x_8 & x_8 & x_8 & -x_8 & \cdots \\ x_9 & -x_5 & -x_{13} & x_1 & -x_{15} & -x_3 & x_{11} & x_7 & -x_7 & -x_{11} & \cdots \\ x_{10} & -x_2 & x_{14} & x_6 & -x_6 & -x_{14} & x_2 & -x_{10} & -x_{10} & x_2 & \cdots \\ x_{11} & -x_1 & x_9 & x_{13} & -x_3 & x_7 & x_{15} & -x_5 & x_5 & -x_{15} & \cdots \\ x_{12} & -x_4 & x_4 & -x_{12} & -x_{12} & x_4 & -x_4 & x_{12} & x_{12} & -x_4 & \cdots \\ x_{13} & -x_7 & x_1 & -x_5 & x_{11} & x_{15} & -x_9 & x_3 & -x_3 & x_9 & \cdots \\ x_{14} & -x_{10} & x_6 & -x_2 & x_2 & -x_6 & x_{10} & -x_{14} & -x_{14} & x_{10} & \cdots \\ x_{15} & -x_{13} & x_{11} & -x_9 & x_7 & -x_5 & x_3 & -x_1 & x_1 & -x_3 & \cdots \end{bmatrix} \quad (8)$$

part,  $T_{8o}$ . Designing an order-16 ICT is equivalent to designing  $T_{8e}$  and  $T_{8o}$ , and the orthogonality of each part is necessary and sufficient for the orthogonality of the order-16 ICT

$$T_{8e} = \begin{bmatrix} x_0 & x_0 \\ x_2 & x_6 & x_{10} & x_{14} & -x_{14} & -x_{10} & -x_6 & -x_2 \\ x_4 & x_{12} & -x_{12} & -x_4 & -x_4 & -x_{12} & x_{12} & x_4 \\ x_6 & -x_{14} & -x_2 & -x_{10} & x_{10} & x_2 & x_{14} & -x_6 \\ x_8 & -x_8 & -x_8 & x_8 & x_8 & -x_8 & -x_8 & x_8 \\ x_{10} & -x_2 & x_{14} & x_6 & -x_6 & -x_{14} & x_2 & -x_{10} \\ x_{12} & -x_4 & x_4 & -x_{12} & -x_{12} & x_4 & -x_4 & x_{12} \\ x_{14} & -x_{10} & x_6 & -x_2 & x_2 & -x_6 & x_{10} & -x_{14} \end{bmatrix} \quad (9)$$

$$T_{8o} = \begin{bmatrix} x_1 & x_3 & x_5 & x_7 & x_9 & x_{11} & x_{13} & x_{15} \\ x_3 & x_9 & x_{15} & -x_{11} & -x_5 & -x_1 & -x_7 & -x_{13} \\ x_5 & x_{15} & -x_7 & -x_3 & -x_{13} & x_9 & x_1 & x_{11} \\ x_7 & -x_{11} & -x_3 & x_{15} & x_1 & x_{13} & -x_5 & -x_9 \\ x_9 & -x_5 & -x_{13} & x_1 & -x_{15} & -x_3 & x_{11} & x_7 \\ x_{11} & -x_1 & x_9 & x_{13} & -x_3 & x_7 & x_{15} & -x_5 \\ x_{13} & -x_7 & x_1 & -x_5 & x_{11} & x_{15} & -x_9 & x_3 \\ x_{15} & -x_{13} & x_{11} & -x_9 & x_7 & -x_5 & x_3 & -x_1 \end{bmatrix}. \quad (10)$$

$T_{8e}$  is an order-8 ICT. For the compatibility, it is designed to be the scaled version of the transform matrix of the order-8 ICT in H.264 or AVS, according to whichever standard it is integrated into. The scaling factor is 4. In other words, the element set of the even part,  $\{x_0, x_2, \dots, x_{12}, x_{14}\}$ , is equal to  $\{32, 40, 40, 36, 32, 24, 16, 8\}$  and  $\{32, 48, 32, 40, 32, 24, 16, 12\}$ , as proposed to AVS and H.264, respectively.

The possible element sets of an orthogonal  $T_{8o}$ , i.e.,  $x_1, x_3, \dots, x_{13}, x_{15}$ , are the solutions of a set of three quadratic equations [21]. The solutions have relatively large magnitudes, i.e., represented by at least 6 bits, and those with small DCT distortion [7] have even larger magnitudes that increase the computational complexity significantly. In this paper, NICT is proposed for the freedom of making trade-off between complexity and performance at the expense of orthogonality. Except orthogonality, NICT preserves all the advantages of ICT, such as bit-exact implementation and the fast algorithm.

Based on the property of orthogonal transforms, if there is no quantization in the frequency domain, signals can be reconstructed perfectly; otherwise, the average variance of the reconstruction error equals that of the quantization error [11]. For a nonorthogonal transform, the average variance of the reconstruction error is larger than that of the quantization error, and even though there is no quantization, the reconstruction is imperfect. Hence, we analyze the transform error introduced by the nonorthogonality.

First of all, we discuss the reconstruction error without quantization. Let vector  $x$  be a sample from a 1-D, zero-mean, unit-variance first-order Markov process with length  $N$  and adjacent element correlation  $\rho$ .  $x$  is transformed by nonorthogonal  $T$ , and the vector  $\theta$  is obtained in the transform domain, i.e.,  $\theta = Tx$ . Let the reconstructed vector be  $y$ , where  $y = T^T\theta = T^T Tx$ . Obviously,  $T^T T$  is not equal to the identity matrix  $I$ , and the difference between  $T^T T$  and  $I$  is denoted as  $E_r$ , i.e.,  $E_r = T^T T - I$ . The average variance of the

reconstruction error  $\sigma_{r0}^2$  is expressed as

$$\begin{aligned} \sigma_{r0}^2 &= \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{E}[(x(k) - y(k))^2] = \frac{1}{N} \mathcal{E}[(x - y)^T (x - y)] \\ &= \frac{1}{N} \mathcal{E}[(E_r x)^T (E_r x)] = \frac{1}{N} \mathcal{E}[x^T E_r^T E_r x]. \end{aligned} \quad (11)$$

Then, we denote  $M = E_r^T E_r$ , and the deduction continues

$$\begin{aligned} \sigma_{r0}^2 &= \frac{1}{N} \mathcal{E}[x^T M x] = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} M(k, j) \mathcal{E}[x(j)x(k)] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} M(k, j) R(j - k) \\ &\leq \frac{1}{N} \sigma_x^2 \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} M(k, j) \end{aligned} \quad (12)$$

where  $R(\cdot)$  and  $\sigma_x^2$  are the autocorrelation function and the variance of the source, respectively. As the autocorrelation is less than or equal to the variance, the upper bound of  $\sigma_{r0}^2$  for a given NICT is shown in (12). Hence, minimizing  $\sigma_{r0}^2$  is equivalent to minimizing the term  $\sum_{k=0}^{N-1} \sum_{j=0}^{N-1} M(k, j)$  in (12). Table III gives some examples of  $\sigma_{r0}^2$  in the design of NICT with  $\sigma_x^2$  equal to 1.0, and it will be explained later.

Then, we take the quantization into consideration and show the relationship of the average variances of the reconstruction error  $\sigma_r^2$  and the quantization error  $\sigma_q^2$ . Let  $u$  be the quantized version of  $\theta$  and  $y_r$  be the reconstructed vector, i.e.,  $y_r = T^T u$ . The quantization error is denoted as  $q$ , i.e.,  $q = \theta - u$ . The average variance of  $\sigma_r^2$  is expressed by (13)

$$\begin{aligned} \sigma_r^2 &= \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{E}[(x(k) - y_r(k))^2] \\ &= \frac{1}{N} \mathcal{E}[(x - y_r)^T (x - y_r)] \\ &= \frac{1}{N} \mathcal{E}[(T^{-1}\theta - T^T u)^T (T^{-1}\theta - T^T u)]. \end{aligned} \quad (13)$$

Since  $T^{-1}$  is not equal to  $T^T$  with the nonorthogonal  $T$ , we use  $(T^T - T_{er})$  to represent  $T^{-1}$ , and substitute  $T^{-1}$  into (13) with  $(T^T - T_{er})$ . Then we get (14)

$$\begin{aligned} \sigma_r^2 &= \frac{1}{N} \mathcal{E}[(T^T q - T_{er}\theta)^T (T^T q - T_{er}\theta)] \\ &= \frac{1}{N} \mathcal{E}[q^T T T^T q - 2q^T T T_{er}\theta + (T_{er}\theta)^T T_{er}\theta] \\ &= \frac{1}{N} \mathcal{E}[q^T (I + E_r)q - 2q^T T (T^T - T^{-1})\theta \\ &\quad + (T^T\theta - T^{-1}\theta)^T (T^T\theta - T^{-1}\theta)] \\ &= \frac{1}{N} \mathcal{E}[q^T q + q^T E_r q - 2q^T E_r\theta + (y - x)^T (y - x)] \\ &= \sigma_q^2 + \sigma_{r0}^2 + \frac{1}{N} \mathcal{E}[q^T E_r q] - \frac{2}{N} \mathcal{E}[q^T E_r\theta]. \end{aligned} \quad (14)$$

The third and fourth terms in (14) are both equal to zero, because the autocorrelation of  $q$  and the cross-correlation of  $q$  and  $\theta$  are both zero as shown by Widrow *et al.* [22]. Therefore, the relationship between  $\sigma_r^2$  and  $\sigma_q^2$  is shown in (15)

$$\sigma_r^2 = \sigma_q^2 + \sigma_{r0}^2. \quad (15)$$

TABLE III  
COMPARISON OF SETS OF MATRIX ELEMENTS

$x_1$	$x_3$	$x_5$	$x_7$	$x_9$	$x_{11}$	$x_{13}$	$x_{15}$	DCT Distortion (%)	Upper Bound of $\sigma_{r0}^2$ ( $10^{-7}$ )
42	38	37	32	22	19	10	4	12.90	0
62	61	49	47	37	31	21	5	10.38	0
94	93	73	70	58	51	26	6	15.71	0
120	114	103	94	68	57	34	14	8.77	0
120	108	104	85	69	52	32	2	11.64	0
28	27	23	21	17	14	8	2	9.04	10.85
29	28	26	22	20	13	10	2	6.92	8.36
38	36	35	29	25	18	9	3	8.53	7.61
39	37	35	29	26	18	11	2	7.36	7.03
40	39	33	31	24	19	14	4	8.70	18.53
40	38	35	31	24	19	11	4	3.74	6.54

Clearly, the average variance of the reconstruction error  $\sigma_r^2$  is always  $\sigma_{r0}^2$  larger than the quantization error variance  $\sigma_q^2$ , no matter the transform coefficients are quantized or not, whereas with an orthogonal transform,  $\sigma_r^2$  is equal to  $\sigma_q^2$ .

The key of designing an NICT is the balance among  $\sigma_{r0}^2$ , the approximation of the DCT, and the magnitudes of the matrix elements. Table III gives 11 element sets of  $T_{8o}$  in (10). Their DCT distortions [7] and upper bounds of  $\sigma_{r0}^2$  in (12) shown in the table are of  $T_{8o}$  instead of  $T_{16}$  in (8). The upper five sets are given by Cham and Chan [21], which lead to orthogonal  $T_{8o}$ , but the magnitudes and DCT distortion of these sets are larger than the rest six sets leading to nonorthogonal  $T_{8o}$ .

Although nonorthogonal, the bottom six element sets are all with negligible  $\sigma_{r0}^2$ , and specially, the one in the bottom row has the minimum DCT distortion and  $\sigma_{r0}^2$ . Another advantage of this element set is that its norm is close to those of the even part  $T_{8e}$  in (9) and therefore is suitable for PIT implementation, which is required when integrated into the AVS platform. Hence, the vector in the bottom row is used for the element set of the odd part, and the performance will be verified in Section VI.

### C. The Proposed Order-16 MICT

The MICT transform matrix is obtained by modifying the structure of the order-16 DCT matrix using the principle of dyadic symmetry. The basis vectors of the order-16 MICT are inherently orthogonal no matter what the elements are, so compared with the elements in the NICT transform matrix, even smaller magnitudes can be selected. The even part of the general transform matrix remains the same as the  $T_{8e}$  in (9), while the odd part is changed, shown as (16)

$$M_{8o} = \begin{bmatrix} x_1 & x_3 & x_5 & x_7 & x_9 & x_{11} & x_{13} & x_{15} \\ x_9 & x_{11} & x_{13} & x_{15} & -x_1 & -x_3 & -x_5 & -x_7 \\ x_5 & x_7 & -x_1 & -x_3 & -x_{13} & -x_{15} & x_9 & x_{11} \\ x_{15} & x_{13} & -x_{11} & -x_9 & x_7 & x_5 & -x_3 & -x_1 \\ x_{13} & -x_{15} & -x_9 & x_{11} & x_5 & -x_7 & -x_1 & x_3 \\ x_3 & -x_1 & -x_7 & x_5 & -x_{11} & x_9 & x_{15} & -x_{13} \\ x_7 & -x_5 & x_3 & -x_1 & -x_{15} & x_{13} & -x_{11} & x_9 \\ x_{11} & -x_9 & x_{15} & -x_{13} & x_3 & -x_1 & x_7 & -x_5 \end{bmatrix}. \quad (16)$$

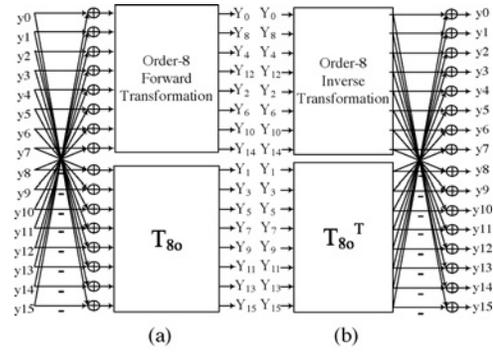


Fig. 3. Compatible structures for order-8 and order-16 (a) forward transformation and (b) inverse transformation.

With the even part  $T_{8e}$  in (9) and the odd part  $M_{8o}$  in (16), the first three basis vectors of the order-16 MICT transform matrix resemble those of the DCT and represent low frequency components. So the MICT shows good energy compaction capability especially for the regions in HD videos, where pixels are higher correlated. Though the dyadic symmetries of most basis vectors in the odd part of the transform matrix are different from those in the DCT matrix, the MICT will not bring significant performance penalty, if the elements are well-selected, as evident in Section VI.

Since the elements in the transform matrices are selected without orthogonality constraint, it is free to make a trade-off between the performance and the magnitudes of the elements, i.e., the computational complexity. In this paper, the magnitudes of elements are represented by only 3 to 4 bits and 16-bit integer implementation can be developed easily. In detail, the even part is selected to be the same as the order-8 ICT in AVS or H.264, according to whichever it is integrated into. For the odd part, there are three considerations for elements' values. First, the magnitudes should be comparable to those of the even part. Second, the waveform of the second basis of the  $16 \times 16$  MICT transform matrix should resemble that of the DCT. Third, it should be suitable for the design of fast algorithm, which will be illustrated in Section IV-B. Taking all these constraints into consideration,  $\{x_1, x_3, \dots, x_{13}, x_{15}\}$  is designed to be  $\{11, 11, 11, 9, 8, 6, 4, 1\}$ .

## IV. COMPLEXITY REDUCTION

This section introduces the research efforts we have made for complexity reduction. First of all, the 2-D order-16 integer transforms are compatible with the 2-D order-8 ICT in AVS EP or H.264 HP, according to whichever standard they are integrated into. Furthermore, the fast algorithm for the 2-D order-16 MICT proposed in Section III-C is developed and extended to a general approach.

### A. Compatibility with the Order-8 ICT

It is natural that the order-16 DCT is compatible with the order-8 one. However, it is not always true for integer transforms. Bossen [23] points out that the compatibility of order-8 and order-4 ICTs can be achieved by taking the order-4 ICT as the even part of the order-8 one. In this paper, the order-

16 integer transforms use the order-8 ICTs in the standards or their scaled version as the even part for the compatibility.

First, the compatible transformation is presented. Fig. 3(a) shows the flow diagram of the forward transformation, applicable for both types of order-16 integer transforms. The bottom-right and upper-right blocks complete the function of matrix multiplication with the odd and even parts of  $T_{16}$ , respectively. In case of the order-16 MICT, where  $T_{8e}$  is exactly the same as the order-8 ICT, the upper-right block may be reused directly for the order-16 forward transformation. In the other case of the order-16 NICT, where  $T_{8e}$  is equal to the order-8 ICT scaled by a factor 4, the output of upper-right block should be left shifted 2 bits before they are reused for the forward transformation of the order-16 NICT. Therefore, a module specially for the order-8 forward transformation is saved, and the output data from the order-8 forward transformation can be reused by the order-16 one. For the inverse transformation, the compatibility can be achieved in a similar fashion, as shown in Fig. 3(b).

Based on the compatibility of transform matrix, the inherent relationship between the scaling matrices,  $R_{8 \times 8}$  and  $R_{16 \times 16}$ , is derived as (17), according to (4) and (5)

$$R_{8 \times 8}(i, j) \times 2^M = R_{16 \times 16}(2i, 2j), \quad 0 \leq i, j < 8 \quad (17)$$

where  $M$  equals 5 and 1 for the case of the NICT and the MICT, respectively. When the divisions in (3) are approximated by integer multiplication and right shift as shown in (6), the relationship between the scaling matrix  $P_{8 \times 8}$  and  $P_{16 \times 16}$  for the scaling is

$$P_{8 \times 8}(i, j) = P_{16 \times 16}(2i, 2j) \times 2^K, \quad 0 \leq i, j < 8 \quad (18)$$

where  $K$  equals 5 and 1 for the case of the NICT and the MICT, respectively. When the order-16 integer transforms are implemented as PIT to be compatible with the order-8 PIT in AVS, (18) is also true for the combined scaling with  $K$  doubled. Interested readers may refer to [24] for more details. Equation (18) indicates that  $P_{8 \times 8}(i, j)$  is  $2^K$  times larger than  $P_{16 \times 16}(2i, 2j)$  if  $N$ , the number of bits for right shift in (6), is the same for the order-16 and the order-8 transforms. The difference between  $P_{8 \times 8}(i, j)$  and  $P_{16 \times 16}(2i, 2j)$  can be easily sorted out by shift operations. In other words,  $P_{16 \times 16}(2i, 2j)$ ,  $0 \leq i, j < 8$ , may be used for the combined scaling of the order-8 ICT but  $N$  in (6) for  $8 \times 8$  scaling is  $K$  bits less than that for the  $16 \times 16$  scaling. Hence, the memory for storing an  $8 \times 8$  scaling matrix is saved, thus achieving the compatibility of scaling matrix.

### B. Fast Algorithm for the Order-16 MICT

ICT and NICT have fast algorithms, which can be developed in a similar way for the DCT. However, fast algorithms for the MICTs are not guaranteed, and little research effort has been devoted to it. We propose a universal approach to developing fast algorithms for the integer transforms with different structures from the DCT matrix, including the MICT. Due to the separability of a 2-D transform and the similarity of the forward and inverse transformation flows, we focus on the fast algorithm for 1-D order-16 transformation in this section.

The eight butterflies in the left part of Fig. 3(a) can be easily developed as the first stage, which exploits the symmetries with respect to the line in (8). The upper-right block completes the function of matrix multiplication with the even part, which is the same as the order-8 ICTs in the standards in this paper. Fortunately, both order-8 ICTs in AVS and H.264 have efficient fast algorithms, which are borrowed here. For a more general case that the even part of the MICT has different structure with  $T_{8e}$  in (9), the fast algorithms can be developed in the same way for  $M_{8o}$  as introduced below.

The bottom-right block in Fig. 3(a), completing the function of matrix multiplication with the odd part, has a quite different structure from the odd part of the order-16 DCT matrix. So we cannot decompose it to butterfly operations just as we do for  $T_{8e}$ . Instead, we represent  $M_{8o}$  as the product of three  $8 \times 8$  matrices, as shown in (19)

$$M_{8o} = M_1 \times M_2 \times M_3. \quad (19)$$

There are some considerations for the three matrices. First, they should contain integers only. Second, the integers should be very small such that only additions and shifts are involved and multiplications are avoided in fast algorithms. Last but not least, the matrices should be sparse, which means that many zeros appear in the matrices.

Obviously, row vectors in  $M_{8o}$  are orthogonal and have the same length, where the length of a row vector  $a$  is defined as  $a \times a^T$ . We denote the length of every row vector of  $M_{8o}$  as  $n_o$ , and clearly  $|\det(M_{8o})|$  is equal to  $n_o^4$ , where  $\det(\cdot)$  and  $|\cdot|$  mean determinant and absolute value, respectively. To simplify the problem, we assume  $M_1$ ,  $M_2$ , and  $M_3$  in (19) are also row-orthogonal and the row vectors have the same length in each matrix, denoted as  $n_1$ ,  $n_2$ , and  $n_3$ , respectively. Similarly,  $|\det(M_1)|$ ,  $|\det(M_2)|$ , and  $|\det(M_3)|$  are equal to  $n_1^4$ ,  $n_2^4$ , and  $n_3^4$ , respectively. If (19) is satisfied, (20) is necessarily true

$$\begin{aligned} |\det(M_{8o})| &= n_o^4 = |\det(M_1)| \times |\det(M_2)| \times |\det(M_3)| \\ &= n_1^4 \times n_2^4 \times n_3^4. \end{aligned} \quad (20)$$

Therefore

$$n_o = n_1 \times n_2 \times n_3. \quad (21)$$

So, when designing an order-16 MICT with a fast algorithm, we should make sure that  $n_o$  of  $M_{8o}$  is the product of three integers. Otherwise, (19) has no solution under this assumption. With the constraint of (21), the lengths of row vectors in  $M_1$ ,  $M_2$ , and  $M_3$  are much smaller than that in  $M_{8o}$ , which means the elements in the three matrices have very small magnitudes and may also contain many zeros.

Among the three matrices,  $M_3$  will be found first. Both sides of (19) are postmultiplied by the inverse of  $M_3$ , which is equal to  $M_3^T/n_3$ , so (19) changes to (22) as follows:

$$M_{8o} \times M_3^T/n_3 = M_1 \times M_2. \quad (22)$$

It is noticed that  $M_3^T$  may be regarded as a set of eight column vectors and  $M_{8o} \times M_3^T/n_3$  contains only integers. So we first collect a set of column vectors by exhaustive search, in which each column vector,  $b_i$ , satisfies two conditions. One is that the length of  $b_i$  is  $n_3$ , and the other is the elements in  $(M_{8o}b_i/n_3)$  are all integers. If (19) has solutions, we can pick out eight

orthogonal column vectors from the set to form  $M_3^T$  and thus get  $M_3$ . To find  $M_2$ , both sides of (22) are postmultiplied by the inverse of  $M_2$ , i.e.,  $M_2^T/n_2$ , so (22) becomes (23)

$$(M_{8o} \times M_3^T/n_3) \times M_2^T/n_2 = M_1. \quad (23)$$

$M_2$  can be obtained by the similar method of searching  $M_3$ . Finally, with  $M_2$  and  $M_3$  known,  $M_1$  is computed by (23).

For the case of  $M_{8o}$  in this paper, its determinant is  $9.9049 \times 10^{10}$  which is equal to  $561^4$  and the corresponding  $n_o$  is 561. To show that  $n_o$  is a product of three integers as in (21), we let  $n_1$ ,  $n_2$ , and  $n_3$  be 3, 11, and 17 with the order changeable. Then the method introduced above is used to search for  $M_1$ ,  $M_2$ , and  $M_3$ , and (24), shown at the bottom of the page, shows one of the solutions of (19). The magnitudes of elements in (24) are so small that the fast algorithm can be implemented by using only additions and shifts. Fig. 4 shows the flow diagram of the forward transformation of the order-16 MICT. The shifts are not indicated in the bottom-right block for a clearer view.

## V. INTEGRATION OF THE 2-D ORDER-16 INTEGER TRANSFORMS INTO THE STANDARDS

In this paper, the proposed two types of 2-D order-16 integer transforms are integrated into AVS EP [5] and H.264 HP [3], respectively, and used adaptively as an alternate to the 2-D order-8 ICTs. For H.264, the 2-D order-4 ICT is not included in the VBT scheme, although it exists in the HP. That is because the 2-D order-4 ICT does not contribute much to the efficiency of HD video coding, as verified in Section VI. The selection of transform size is based on the MB-level R-D cost [17]. The transform with lower cost is selected and a 1-bit binary signal is transmitted in MB header for the indication.

For AVS EP, the largest block size for intra prediction is  $8 \times 8$ . To apply the 2-D order-16 integer transforms to intra-coded MB, the  $16 \times 16$  intra prediction is employed, which is simply extended from the five prediction modes in AVS. The reference pixels are not used for prediction directly, but filtered by a 3-tap low-pass filter  $[1 \ 2 \ 1]/4$  at first. Otherwise, visible artifacts would be introduced due to large block size intra prediction [25], [26].

In AVS EP and H.264 HP, the  $8 \times 8$  residual blocks are coded by CABAC [3], [5]. Their arithmetic coding engines can code sources with various statistics by using many automatically updated probability models and therefore remain unchanged in this paper. Two additional sets of probability models specially for coding the  $16 \times 16$  residual blocks are designed for AVS and H.264, respectively, to drive the arithmetic coding engine.

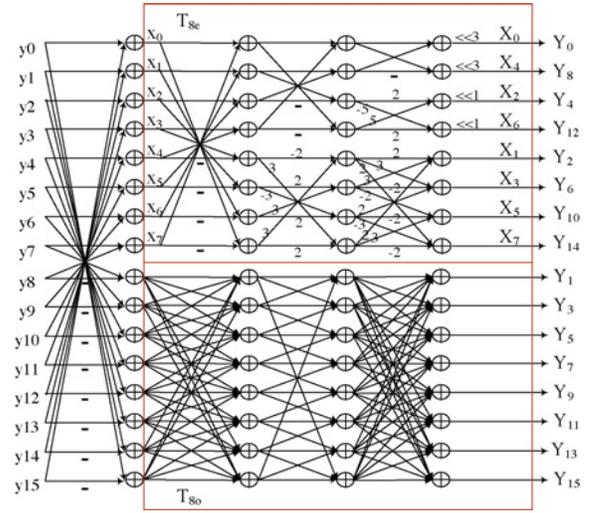


Fig. 4. Flow diagram of the order-16 forward transformation.

CBP is a 6-bit integer in AVS and H.264, of which the four least significant bits (LSB) indicate whether the four  $8 \times 8$  luminance blocks contain nonzero coefficients, respectively. In this paper, if an MB is coded by the 2-D order-16 transform, CBP becomes a 3-bit integer with the LSB indicating whether the  $16 \times 16$  luminance block contains nonzero coefficients.

## VI. EXPERIMENTAL RESULTS

### A. Transform Coding Gain

An important measure for the evaluation of the compression performance is the transform coding gain  $G_{TC}$  [11], which weights the reconstruction error variance of quantized transform coding  $\sigma_{TC}^2$  and the corresponding reconstruction error variance of PCM coding  $\sigma_{PCM}^2$

$$G_{TC} = \frac{\sigma_{PCM}^2}{\sigma_{TC}^2}. \quad (25)$$

Under the assumptions of optimum quantization and bit allocation at the same overall rate,  $G_{TC}$  is the ratio of arithmetic to geometric means of the variances of transform coefficients  $\sigma_{\theta k}^2$  ( $0 \leq k < N$ ), as shown in (26)

$$G_{TC} = \frac{\frac{1}{N} \sum_{k=0}^{N-1} \sigma_{\theta k}^2}{\left(\prod_{k=0}^{N-1} \sigma_{\theta k}^2\right)^{\frac{1}{N}}}. \quad (26)$$

Suppose the source is 1-D zero-mean, unit-variance first-order Markov processes with adjacent element correlation  $\rho$

$$M_{8o} = M_1 \times M_2 \times M_3 = \begin{bmatrix} -2 & 0 & 1 & -1 & -1 & 3 & -1 & 0 \\ 3 & -1 & 1 & 1 & 0 & 2 & 0 & 1 \\ -1 & -3 & 1 & 0 & 1 & 0 & 2 & -1 \\ 0 & 1 & 0 & 1 & 3 & 1 & -1 & -2 \\ 1 & -1 & -3 & -2 & 0 & 1 & 0 & -1 \\ 1 & 1 & 1 & 0 & -2 & 0 & 1 & -3 \\ 0 & -2 & 0 & 1 & -1 & -1 & -3 & -1 \\ -1 & 0 & -2 & 3 & -1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -2 & 0 & -1 & 0 & 2 & -1 & 1 \\ 0 & -1 & 0 & 2 & 1 & -1 & 0 & 2 \\ 0 & -2 & 1 & 1 & 0 & 0 & 1 & -2 \\ -2 & 0 & -1 & 0 & 2 & 0 & -1 & -1 \\ -1 & 0 & 2 & 0 & -1 & -1 & -2 & 0 \\ -2 & 0 & 1 & -1 & 0 & 0 & 2 & 1 \\ -1 & 1 & 0 & 2 & -1 & 2 & 0 & 0 \\ -1 & -1 & -2 & 0 & -2 & -1 & 0 & 0 \end{bmatrix} \quad (24)$$

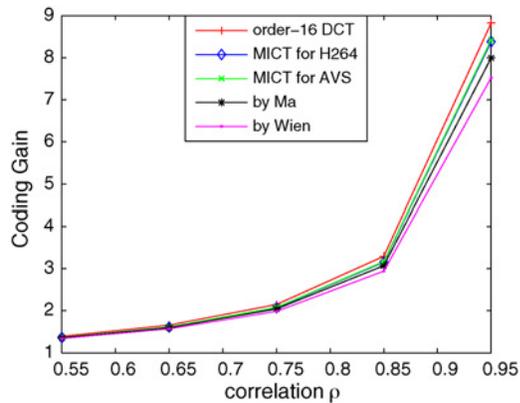


Fig. 5. Comparing coding gain of orthogonal transforms.

ranging from 0.55 to 0.95, which is typical for the prediction errors of HD videos (see Table II). Fig. 5 shows the coding gains of the order-16 DCT, the proposed MICTs for AVS and H.264, respectively, and the two MICTs proposed by Wien [7] and Ma [10], respectively. DCT has the highest coding gain, no matter what the value of  $\rho$  is. The coding gains of the proposed two MICTs are too close to be distinguished from each other, and are higher than those of the MICTs proposed by Ma and Wien, especially when  $\rho$  approaches 1.0.

### B. Reconstruction Error of Nonorthogonal Transforms

In the case of nonorthogonal transforms, even with optimal quantization and bit allocation, (26) cannot be derived to substitute (25), because  $\sigma_{PCM}^2$  depends on quantization only, whereas  $\sigma_{TC}^2$  depends not only on quantization but also on the variance of the source. Therefore, we cannot evaluate the performance of the proposed NICTs by (26) and instead, we have to implement the transforms and quantization and compare the reconstruction errors thus introduced.

Seven order-16 transforms are tested, including the order-16 DCT, the proposed two NICTs and the four ICTs, of which the even part is the order-8 ICT in H.264 and the element sets of the odd part are shown in the first four rows of Table III. These transforms are applied to the 1-D zero-mean, unit-variance first-order Markov processes with adjacent element correlation  $\rho$  ranging from 0.55 to 0.95. Before inverse transformation, the transform coefficients are zonal filtered [11], which means the bit rate is reduced by simply retaining the first  $n$  ( $n < 16$ ) transform coefficients. The MSE between the source and the reconstructed value are calculated.

Fig. 6 compares the MSE introduced by 4:1, 4:2, and 4:3 zonal filters. In detail, the three zonal filters retain the first 4, 8, and 12 transform coefficients respectively, and set all the others to zero. For a clear view, the curves in Fig. 6 present the difference MSE compared with that of the order-16 DCT. We notice that the four ICTs produce larger MSE than the proposed two NICT, although orthogonal. Therefore, the performance of an integer transform is greatly determined by the approximation of the exact DCT rather than the orthogonality, if  $\sigma_{r0}^2$  is negligible compared with  $\sigma_q^2$  in (15).

TABLE IV  
TEST CONDITIONS

Platform	RM62b (AVS), JM11 (H.264)
Sequence structure	IBBPBBP...
Intra frame period	0.5 s
Entropy coding	Arithmetic coding
Fast motion estimation	on
Deblocking filter	on
R-D optimization	on
QP	fixed (27, 30, 35, 40, AVS)
	fixed (20, 24, 28, 32, H.264)
Rate control	off
Reference frame	2 (AVS), 5 (H.264)
Search range	$\pm 32$
Frame number	60

TABLE V  
EXPERIMENTAL RESULTS ON THE AVS PLATFORM

HD Sequence	Proposed NICT		Proposed MICT	
	PSNR Gain (dB)	Bit Saving (%)	PSNR Gain (dB)	Bit Saving (%)
<i>Bigship</i>	0.167	-5.98	0.123	-4.65
<i>City</i>	0.170	-5.69	0.123	-4.39
<i>Crew</i>	0.214	-9.31	0.219	-9.81
<i>Optis</i>	0.209	-7.65	0.136	-5.21
<i>Raven</i>	0.132	-4.14	0.087	-2.83
<i>Sheriff</i>	0.162	-5.66	0.086	-3.19
<i>Pedestrian</i>	0.227	-6.90	0.223	-7.12
<i>Riverbed</i>	0.471	-8.86	0.235	-4.77
<i>RushHour</i>	0.198	-7.07	0.190	-7.45
<i>Station</i>	0.195	-8.74	0.116	-4.02
<i>Sunflower</i>	0.158	-4.41	0.151	-4.41
<i>Tractor</i>	0.192	-5.49	0.087	-2.57
Average	0.21	-6.66	0.15	-5.04

### C. Objective Evaluation

The proposed two types of 2-D order-16 integer transforms were, respectively, integrated into the RM62b platform for AVS EP and the JM11 platform for H.264 HP, using 16-bit integer arithmetic. The implementation problems are solved in Section V. The test conditions are listed in Table IV. Tables V and VI show performance gains, compared to AVS EP using a 2-D order-8 ICT and H.264 HP using both 2-D order-4 and order-8 ICTs, respectively. The improvement is measured by PSNR gain for equal bit rate or by bit rate saving at the same PSNR, using the method in [27].

With the NICT, the improvements are all greater than 0.1 dB and more than 0.2 dB on average, whilst for the best case of *Riverbed*, the gain is up to 0.47 dB on the AVS platform and 0.48 dB on the H.264 platform. The sequence *Riverbed* has smooth textures and global motions, so the energy of prediction errors is quite small, no matter using inter or intra prediction. In the transform domain, the prediction errors can be represented by only a few coefficients of the 2-D order-16 NICT, most of which cluster in the low frequency area.

With the MICT, the coding efficiency is improved on an average of around 0.06 dB less than that by NICT on both platforms, but for the sequences with large homogeneous regions or smooth motions, such as *Crew*, *Pedestrian*, *RushHour*,

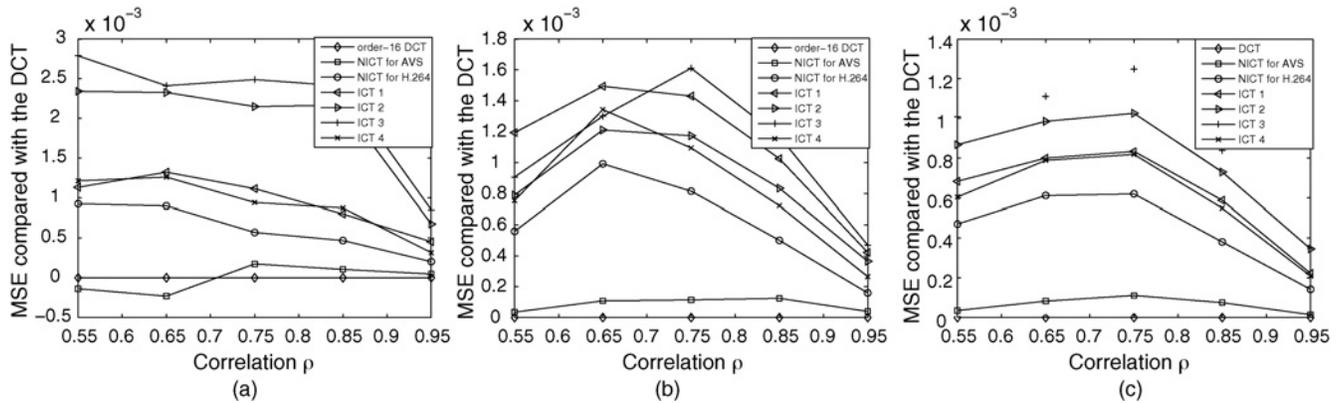


Fig. 6. MSE with (a) 4:1, (b) 4:2, and (c) 4:3 zonal filtering.

TABLE VI  
EXPERIMENTAL RESULTS ON THE H.264 PLATFORM

HD Sequence	Proposed NICT				Proposed MICT			
	Order-4, order-8, order-16		Order-8, order-16		Order-4, order-8, order-16		Order-8, order-16	
	PSNR Gain (dB)	Bit Saving (%)	PSNR Gain (dB)	Bit Saving (%)	PSNR Gain (dB)	Bit Saving (%)	PSNR Gain (dB)	Bit Saving (%)
<i>BigShip</i>	0.14	-5.09	0.10	-3.76	0.12	-4.51	0.08	-3.13
<i>City</i>	0.17	-5.29	0.13	-4.18	0.13	-4.27	0.10	-3.21
<i>Crew</i>	0.18	-7.71	0.19	-8.26	0.17	-7.49	0.18	-7.98
<i>Optis</i>	0.28	-9.47	0.28	-9.46	0.24	-8.33	0.23	-7.96
<i>Raven</i>	0.33	-10.61	0.30	-9.87	0.27	-8.70	0.23	-7.48
<i>Sheriff</i>	0.16	-5.08	0.14	-4.56	0.12	-3.81	0.09	-3.02
<i>Pedestrian</i>	0.18	-6.77	0.20	-7.59	0.14	-5.59	0.15	-5.92
<i>Riverbed</i>	0.40	-8.55	0.48	-10.20	0.10	-2.28	0.14	-3.16
<i>RushHour</i>	0.15	-7.41	0.14	-6.95	0.11	-5.90	0.09	-4.82
<i>Station</i>	0.23	-9.48	0.23	-10.09	0.17	-7.04	0.17	-7.55
<i>Sunflower</i>	0.45	-15.16	0.40	-13.92	0.44	-14.96	0.39	-13.50
<i>Tractor</i>	0.30	-8.35	0.27	-7.75	0.22	-6.28	0.16	-4.80
Average	0.25	-8.25	0.24	-8.05	0.19	-6.60	0.17	-6.05

and *Sunflower*, the performance of the MICT is comparable to or even better than the NICT, due to the efficiency of the first three basis vectors. However, for the sequences full of details, such as *Sheriff*, *Station*, and *Tractor*, the gain of the MICT becomes trivial because of the relative inefficiency to compress high frequency components. Overall, the performance gap between the proposed NICT and MICT is small.

Fig. 7(a) and (b) show the percentage of MBs coded by the 2-D order-16 integer transforms. On average, more than half of the MBs are coded by 2-D order-16 transforms, and for some sequences, such as *Sunflower* and *Station*, the percentage of 2-D order-16 transforms is up to 80%. This implies that 2-D order-16 integer transforms are very useful in HD video coding. There is a tendency that the higher the bit rate is, the less the 2-D order-16 integer transforms are used. That is because at high bit rates, many high frequency coefficients survive the quantization and high-cost coefficients, i.e., a (*level*, *run*) pair with large *run*, have to be coded due to the large block size of  $16 \times 16$ . Hence, 2-D order-8 ICT is more likely to be selected.

We also study the VBT scheme using three transforms, 2-D order-4, order-8, and order-16 transforms, on the H.264

platform. As shown in Table VI, the performance gap of the two VBT schemes with and without the 2-D order-4 ICT is marginal. The same conclusion can also be drawn by Fig. 7(c) and (d), where the percentages of MBs coded by 2-D order-4, order-8, and order-16 transforms are compared on the H.264 platform. The percentage of using 2-D order-4 ICT is very small as we expected. Therefore, in the proposed VBT scheme, the 2-D order-4 ICT in H.264 is removed.

#### D. Subjective Evaluation

Using 2-D order-16 transform can preserve more details and provide better visual quality, especially at low bit rate. Fig. 8 gives two examples for the subjective comparison. The images are all cropped from H.264-coded HD videos with size of  $150 \times 150$  pixels. The indicated PSNR and number of bits are the R-D information of the associated frames. Obviously, the vertical edges of the buildings in *City* and the horizontal edges of the railway sleepers in *Station* are better preserved by the additional 2-D order-16 transforms. However, at high bit rate, the visual quality without using 2-D order-16 transform is good enough and therefore the improved fidelity by using 2-D order-16 transforms is hard to be visually observed.

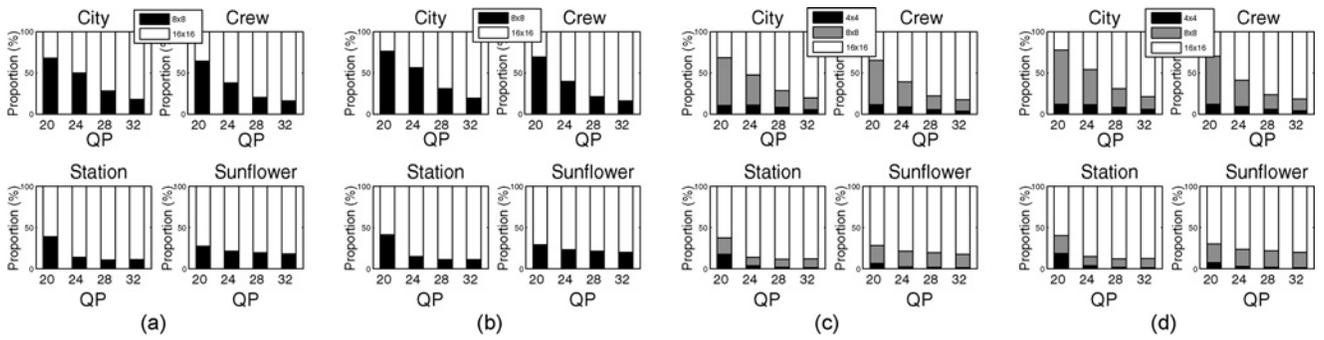


Fig. 7. Proportion of different block size transforms used in H.264 HP. (a) 2-D order-8 ICT and order-16 NICT. (b) 2-D order-8 ICT and order-16 MICT. (c) 2-D order-8 and order-4 ICTs and 2-D order-16 NICT. (d) 2-D order-8 and order-4 ICTs and 2-D order-16 MICT.

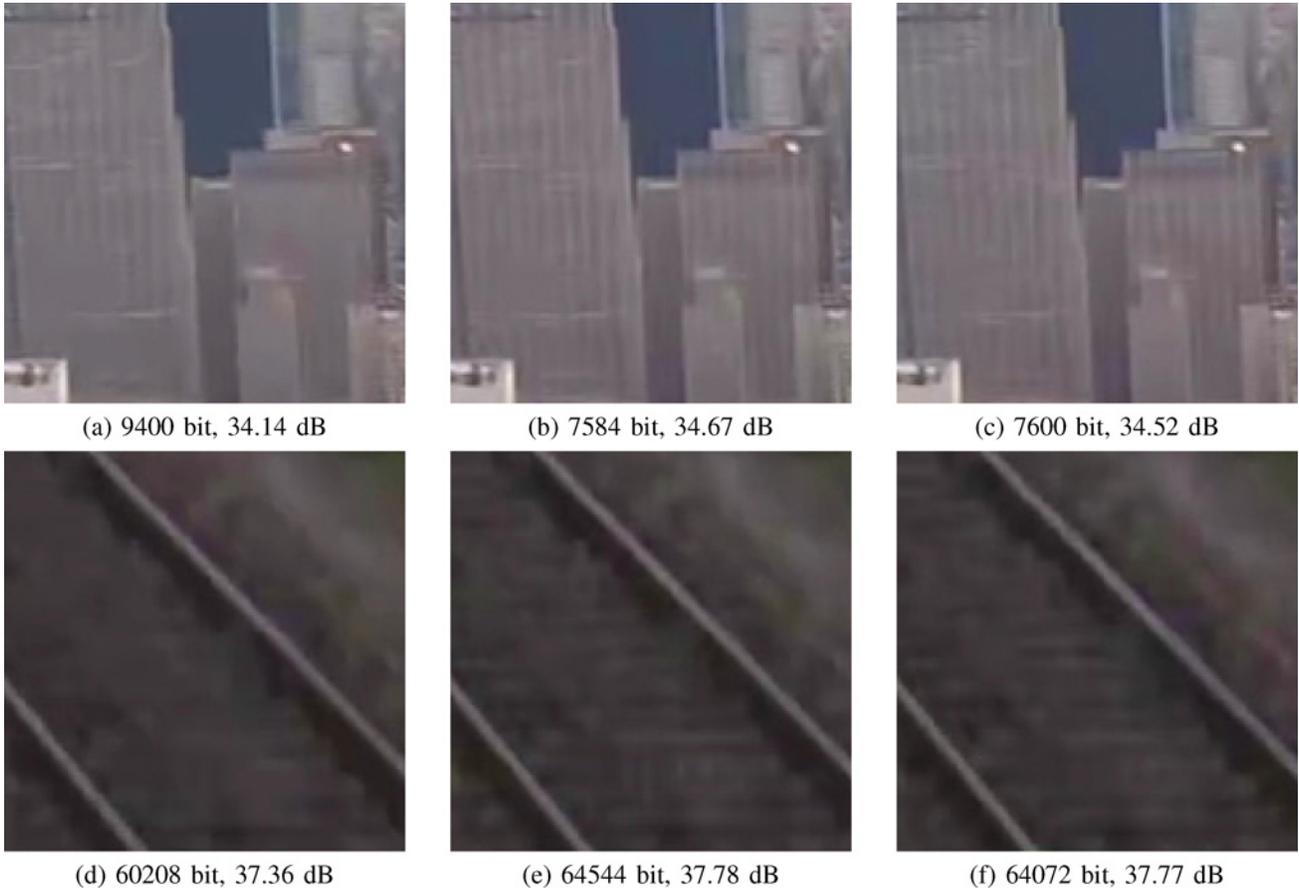


Fig. 8. Images cropped from *City* (720p) and *Station* (1080p) both coded with QP = 32. (a) and (d) are coded using H.264 HP. (b) and (e) are coded using H.264 HP with additional 2-D order-16 NICT. (c) and (f) are coded using H.264 HP with additional 2-D order-16 MICT.

TABLE VII

OPERATIONS OF FAST ALGORITHM AND MATRIX MULTIPLICATION

Operation	Fast MICT	Matrix Multiplication
Addition	150	240
Multiplication	0	256
Shift	32	0

TABLE VIII

EXECUTION TIME OF FAST ALGORITHMS AND MATRIX MULTIPLICATION

	Fast Algorithm (s)	Matrix Multiplication (s)	Saving Time (%)
DCT	0.040	0.568	93%
MICT	0.031	0.282	89%
$M_{8o}$	0.005	0.026	77%

### E. Efficiency of the Fast Algorithm

If the transform is implemented by matrix multiplication, the complexity is high, due to many additions and multiplications introduced by the dot products. The fast algorithm developed

in Section IV-B can implement the transformation process of the MICT by shifts and additions only. Table VII compares the numbers of additions, shifts, and multiplications required by transforming a 16-point vector using the fast algorithm and

matrix multiplication, respectively. Without multiplication, the computational complexity is reduced significantly. However, the number of additions required for the fast algorithm is considerable, because multiplications with small magnitude integers are replaced by combinations of additions and shifts.

The 2-D order-16 MICT was implemented using both fast algorithm and matrix multiplication by the C language. Ten thousand  $16 \times 16$  random blocks were generated, in which the numbers were uniformly distributed from  $-256$  to  $255$ . The total execution time for transforming these blocks by the two methods was recorded, respectively. The PC platform has a 3.2 GHz Intel Pentium 4 CPU and 1.0 GB RAM. Two Windows API functions QueryPerformanceFrequency and QueryPerformanceCounter are used for timing [28], which measure the execution time to nanosecond accuracy. The time comparison is shown in the third row of Table VIII, where using fast algorithm can save about 89% of the computational time.

The efficiency of the fast algorithm for the MICT is compared with that of the fast DCT (FDCT) [29], because there are no publications related to fast algorithms for 2-D order-16 ICT, NICT or MICT. We did the experiment as described above for the case of the DCT, and the time for computing the 2-D order-16 DCT using FDCT and matrix multiplication are recorded, respectively. The second row of Table VIII shows FDCT can save 93% of the computational time compared with using matrix multiplication. The proposed fast algorithm for the 2-D order-16 MICT is 4% less efficient than the FDCT for the DCT because of the different dyadic symmetries in  $M_{8o}$ .

At the same time, we are also concerned with the efficiency of the fast algorithm used for the bottom-right module in Fig. 3(a), which is specially designed for the multiplication with matrix  $M_{8o}$ . Similarly, ten thousand  $8 \times 8$  random blocks were generated, in which the numbers were uniformly distributed from  $-256$  to  $255$ . As shown in the fourth row of Table VIII, the fast algorithm can save almost 77% computational time compared with matrix multiplication. This proves that the fast algorithm for  $M_{8o}$  has lower efficiency than that of the whole 2-D order-16 MICT. If the structure of the even part is different from  $T_{8e}$  in (9) and some existing fast algorithms of 2-D order-8 ICTs cannot be used, the method for decomposing  $M_{8o}$  has to be applied to the even part and the fast algorithm will be about 10% less efficient, but overall, the computational complexity is reduced significantly.

## VII. CONCLUSION

In this paper, 2-D order-16 transforms are proposed to exploit the spatial property of HD videos. Specifically, we propose two types of 2-D order-16 integer transforms, NICT and MICT, both of which allow selecting matrix elements more freely by releasing the orthogonality constraint and can provide comparable performance with that of the DCT. The latter is even simpler. Furthermore, considerable effort has been spent to reduce the complexity, including compatibility with the 2-D order-8 ICT in AVS or H.264 and the development of the fast algorithm. Both 2-D order-16 integer transforms are integrated into AVS EP and H.264 HP,

respectively, and used adaptively as an alternative to the 2-D order-8 ICTs according to local activities. The experimental results show that 2-D order-16 integer transforms can be efficient coding tools especially for HD video coding.

## REFERENCES

- [1] *Generic Coding of Moving Pictures and Associated Audio Information Part 2: Video*, ITU-T Rec. H.262 and ISO/IEC 13818-2 MPEG-2, 1998.
- [2] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, document JVT-G050, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, Mar. 2003, [Online]. Available: <http://ftp3.itu.ch/avarch/jvt-site>
- [3] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, Mar. 2005.
- [4] *Information Technology-Advanced Coding of Audio and Video-Part 2: Video*, GB/T 20090.2-2006 AVS, May 2006.
- [5] AVS Video Group. (2006, Sep.). *Information Technology-Advanced Coding of Audio and Video: Part 2. Video*, AVS-N1318.
- [6] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Boston, MA: Kluwer, 2002.
- [7] *ICT Comparison for Adaptive Block Transforms*, document ITU-T SG16/Q6 VCEG and VCEG-L12, Jan. 2001, [Online]. Available: <http://ftp3.itu.ch/av-arch/video-site>
- [8] *Simplified Adaptive Block 762 Transforms*, document ITU-T SG16/Q6 VCEG, Dec. 2001, [Online]. Available: <http://ftp3.itu.ch/av-arch/video-site/VCEG-O30r1.doc>
- [9] S. Naito and A. Koike, "Efficient coding scheme for super high definition video based on extending H.264 high profile," in *Proc. SPIE Vis. Commun. Image Process.*, vol. 6077. Jan. 2006, pp. 607727-1-607727-8.
- [10] S. Ma and C.-C. Kuo, "High-definition video coding with super-macroblocks," in *Proc. SPIE Vis. Commun. Image Process.*, vol. 6508. Jan. 2007, pp. 650816-1-650816-12.
- [11] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [12] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3032-3044, Dec. 1991.
- [13] *A 16-bit Architecture for H.264, Treating DCT Transform and Quantization*, document ITU-T SG16/Q6 VCEG and VCEG-M16, Apr. 2001, [Online]. Available: <http://ftp3.itu.ch/av-arch/video-site>
- [14] A. Habibi, "Comparison of nth-order DPCM encoder with linear transformation and block quantization techniques," *IEEE Trans. Commun.*, vol. 19, no. 6, pp. 948-956, Dec. 1971.
- [15] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 2001.
- [16] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*. Boston, MA: McGraw-Hill, 2002.
- [17] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3. Oct. 2001, pp. 542-545.
- [18] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, no. 3, pp. 366-406, Mar. 1980.
- [19] W. K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," in *Proc. Inst. Electr. Eng. I: Commun. Speech Vis.*, vol. 136, no. 4, 1989, pp. 276-282.
- [20] C.-X. Zhang, J. Lou, L. Yu, J. Dong, and W. K. Cham, "The technique of pre-scaled integer transform in hybrid video coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1. May 2005, pp. 316-319.
- [21] W. K. Cham and Y. T. Chan, "An order-16 integer cosine transform," *IEEE Trans. Signal Process.*, vol. 39, no. 5, pp. 1205-1208, May 1991.
- [22] B. Widrow, I. Kollar, and M.-C. Liu, "Statistical theory of quantization," *IEEE Trans. Instrum. Meas.*, vol. 45, no. 2, pp. 353-361, Apr. 1996.
- [23] *ABT Cleanup and Complexity Reduction*, document JVT-E087.doc, Oct. 2002, [Online]. Available: <http://ftp3.itu.ch/av-arch/jvt-site>
- [24] J. Dong, J. Lou, C.-X. Zhang, and L. Yu, "A new approach to compatible adaptive block size transforms," in *Proc. SPIE Vis. Commun. Image Process.*, vol. 5960. Jul. 2005, pp. 38-47.
- [25] M. Wien, "Variable block size transforms for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, Jul. 2003.
- [26] *Study of ABT for HDTV Coding*, document JVT-E073r2.doc, Oct. 2002, [Online]. Available: <http://ftp3.itu.ch/av-arch/jvt-site>
- [27] *Calculation of Average PSNR Differences Between RD-curves*, document ITU-T SG16/Q6 VCEG, Apr. 2001, [Online]. Available: <http://ftp3.itu.ch/av-arch/video-site/VCEG-M33.doc>

- [28] Microsoft Developer Network. [Online]. Available: [http://msdn2.microsoft.com/en-us/library/ms674866\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms674866(VS.85).aspx)
- [29] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2. May 1989, pp. 988–991.



**Jie Dong** (S'07) received the B.Eng. and M.Eng. degrees in information engineering from Zhejiang University, Hangzhou, China, in 2002 and 2005, respectively. She is currently working toward the Ph.D. degree in electronic engineering from the Chinese University of Hong Kong, Hong Kong, China. Her research interests include HD video compression and processing.



**King Ngai Ngan** (M'79–SM'91–F'00) received the Ph.D. degree in electrical engineering from Loughborough University of Technology, Leicestershire, U.K.

Currently, he is a Chair Professor at the Department of Electronic Engineering, the Chinese University of Hong Kong, Hong Kong, China, and was previously a Full Professor at Nanyang Technological University, Singapore, and University of Western Australia, Perth. He is the author of three books, five edited volumes, and over 200 refereed technical papers in the areas of image/video coding and communications.

Prof. Ngan is an Associate Editor of the *Journal on Visual Communications and Image Representation*, as well as an Area Editor of the European Association for Signal Processing *Journal of Signal Processing: Image Communication*, and served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and *Journal of Applied Signal Processing*. He chaired a number of prestigious international conferences on video signal processing and communications and served on the advisory and technical committees of numerous professional organizations. He is a general co-chair of the IEEE International Conference on Image Processing, to be held in Hong Kong, in 2010. He is a Fellow of The Institution of Engineering and Technology (U.K.), and Institution of Engineers of Australia, and was an IEEE Distinguished Lecturer during 2006–2007.



**Chi-Keung Fong** (S'07) received the B.Eng. degree in electrical and electronic engineering from The University of Hong Kong, Hong Kong, China, in 1999, and the M.Phil. degree in electronic engineering in 2003, from the Chinese University of Hong Kong, Hong Kong, China, where he is currently working toward the Ph.D. degree.

His research interests include transform coding, pattern matching, and image processing.



**Wai-Kuen Cham** (S'77–M'79–SM'91) graduated from the Chinese University of Hong Kong, Hong Kong, China, in 1979, in electronics. He received the M.Sc. and Ph.D. degrees from Loughborough University of Technology, Leicestershire, U.K., in 1980 and 1983, respectively.

Since May 1985, he has been with the Department of Electronic Engineering, the Chinese University of Hong Kong. His research interests include image coding, image processing, and video coding.