Two-Layer Directional Transform for High Performance Video Coding

Jie Dong, *Member, IEEE,* and King Ngi Ngan, *Fellow, IEEE*

***Abstract*—This paper presents a directional transform scheme for coding interprediction errors in block-based hybrid video coding. It proposes a two-layer transform structure, where the first layer uses discrete wavelet transform to compact the residue energy to the LL band and then the second layer uses 2-D nonseparable directional transforms to deal with the arbitrary edge directions in the four subbands. By doing this, the edges in a macroblock are efficiently compacted to a few coefficients and at the same time the overhead used to indicate the transform directions is affordable. Experimental results show that the proposed scheme provides peak signal-to-noise ratio gain up to 0.46 dB, compared with H.264/AVC High Profile.**

***Index Terms*—AVC, H.264, HEVC, transform.**

I. Introduction

THE EXISTING video coding standards use discrete cosine transform (DCT) or its simplified variants, such as integer cosine transform (ICT) [1], for transform coding. The underlying assumption is that the signal to be transformed can be modeled by a first-order Markov random field with the correlation coefficient approaching 0.95 [2]. This assumption has been verified for images [3]. In video coding, where the input of the transform is the inter or intraprediction errors instead of the original pixels, the assumption is violated. Based on our study [4], the correlation coefficient varies from 0.1 to 0.9, depending on the spatial resolution. The reason is that edges become dominant in a residual frame. In a frame to be coded, the energy in smooth areas is greatly reduced by motion-compensated prediction (MCP), but in edge areas, where the prediction is sensitive to occlusion and geometric distortion, the residue's energy along the edge direction is still strong. When the edges are neither horizontal nor vertical, conventional transforms result in large-magnitude high-frequency coefficients that not only need more bits to code but also introduce large quantization noise at low bit-rates.

Directional transform, which efficiently compacts energy along arbitrary edges, has been a research topic in image coding for many years, and will have more impact on video coding, where efficiently dealing with edges becomes more critical according to the above rationale. The existing directional transforms fall into three categories: discrete wavelet transform (DWT), DCT, and Karhunen–Loeve transform (KLT).

Directional DWTs are mainly proposed for image coding. To resolve the conflict of global transform and local features, directional DWTs incorporate a local directional prediction into the lifting stage, which involves only a few neighboring pixels in signal prediction and updating. Claypoole *et al.* [5] developed a nonlinear lifting structure, in which the order of the filter in the predict step is selected in a way that the filtering will not cross edges. Taubman [6] proposed a nonseparable lifting structure, which allows the filtering direction in the predict step to adapt to arbitrary edge directions. Gerek and Cetin [7] proposed a separable lifting structure with 2-D edge-adaptive prediction, which means the 2-D directional DWT is accomplished by performing two 1-D DWTs in horizontal and vertical directions successively and the prediction filter of each 1-D DWT is supported by diagonal pixels according to the proposed edge direction estimator. However, the 2-D edge-adaptive prediction works for 5/3-tap biorthogonal wavelet filter only. Ding *et al.* extend the directional DWT in [7] by proposing the adaptive directional lifting-based DWT [8], where each 1-D wavelet can be realized in arbitrarily directional prediction by any wavelet filter, including the popular 5/3-tap and 9/7-tap biorthogonal wavelet filters. Subsequently, Chang *et al*. [9] proposed to use quincunx sampling in the same framework of [8]. In [10], directional filter banks are applied to the high-pass subbands transformed by a separable DWT, where the directional filter banks are replaced by diagonally quadrant filter banks plus directional DWT in [11] to reduce the computational complexity. In [12], Liu and Ngan proposed weighted adaptive lifting-based DWT to avoid the prediction/update mismatch in [8] and [9] and to use flexible interpolation directions and filters. The work in [12] is extended and applied to wavelet-based video coding as in [13].

Directional DCTs are developed for the block-based image coding, such as JPEG. The work done by Zeng and Fu [14] incorporates directional information into DCT. The directional DCT is inspired by shape-adaptive DCT [15], adopted in MPEG-4 to code the arbitrary-shaped image segments. The samples in a block are reorganized along the edge directions and then transformed. Although the correlation along the

Manuscript received February 16, 2011; revised June 12, 2011 and September 5, 2011; accepted September 7, 2011. Date of publication October 10, 2011; date of current version April 2, 2012. This work was supported in part by a grant from the Research Grants Council of Hong Kong, under Project CUHK416010. This paper was recommended by Associate Editor A. Vetro.

J. Dong is with the Office of the Chief Technical Officer, InterDigital Communications, San Diego, CA 92121 USA (e-mail: jie.dong@interdigital.com).

K. N. Ngan is with the Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: knngan@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.2011.2171212

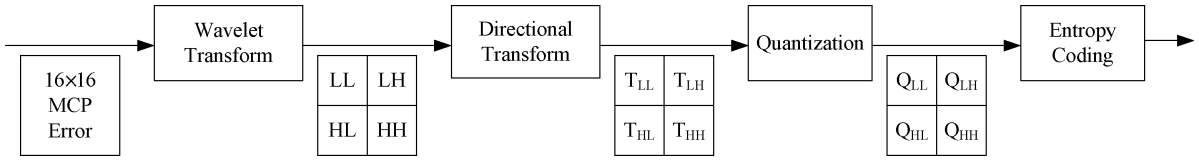


Fig. 1. Block diagram of the proposed two-layer directional transform.

edge is efficiently exploited, the average order of DCT is reduced because of the reorganization, which hurts the overall performance improvement. Similar to [14], the work in [16] and [17] also reorganized the samples in a block along the edge directions. The algorithm in [16] performed only the first direction of transform on the reorganized signal and skips the second direction of transform; the algorithm in [17] performed the second direction of transform only on the DC coefficients of the output of the first direction of transform. Inspired by the fact that 1-D DCT can be factorized into a series of lifting operations, Xu *et al*. proposed a lifting-based directional DCT-like transform [18], which is constructed by directional lifting operations. In a directional lifting operation, the linked orientation of the two involved pixels is consistent with the edge direction. By adapting the directional transform to primary lifting operations, the flexibility of DCT is significantly improved. The directional DCT-like transform can be performed along arbitrary directions in theory and the correlation among neighboring blocks with the similar direction is also exploited.

Recently, Zhu *et al*. proposed using nonseparable KLT in order to study the rate-distortion (R-D) performance upper bound of directional transform [19]. Directional information is introduced to the source model and reflected in the correlation matrix. Then, KLT matrix is constructed by the eigenvectors of the correlation matrix. The experiments on artificial images that have globally consistent edges show that the upper bound of the performance improvement is 15 dB, compared with traditional 2-D DCT. Ye and Karczewicz proposed a series of predefined separable transforms [20] to the intracoding of H.264/AVC, in order to transform the intraprediction error that still has significant directional information; each transform favors one of the intraprediction directions. The transform matrices for horizontal and vertical transforms are constructed by the eigenvectors of the horizontal and vertical correlation matrices, respectively, like the construction of KLT matrix.

Most of the directional transforms reviewed above are proposed for image coding or for transforming intraprediction errors in video coding. The reason for not using them for interprediction errors is that the proportion of the side information for indicating the edge direction becomes relatively large. Although the edges are compacted to fewer coefficients by the directional transforms and less bits are used to coded them, the bit-rate reduction is canceled out by the bit-rate increase for the direction indication. This paper proposes a two-layer transform structure, where the first layer uses 2-D DWT to compact the residue energy to the LL band and then the second layer uses 2-D nonseparable directional transforms to deal with the arbitrary edge directions in four subbands. By doing this, the edges are efficiently compacted to a few coefficients and the overhead used to indicate the transform direction is still affordable. Experimental results show that the proposed scheme provides peak signal-to-noise ratio (PSNR) gain up to 0.46 dB, compared with H.264/AVC High Profile.

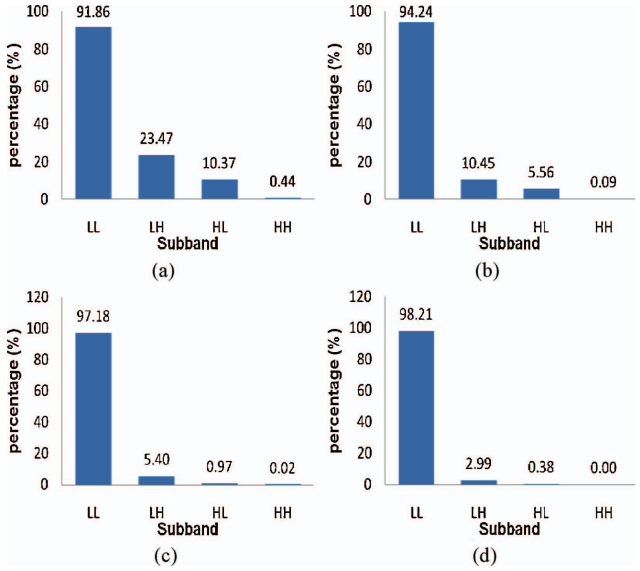


Fig. 2. Given an MB using directional transform, the probabilities of having nonzero coefficients in four subbands, when QPs are equal to (a) 22, (b) 27, (c) 32, and (d) 37, respectively.

The remainder of this paper is organized as follows. Section I presents the two-layer direction transform, including the transform structure, the matrix design, and how it is integrated into H.264/AVC. Experimental results are shown in Section III, followed by the conclusion in Section IV.

II. Two-Layer Directional Transform

A. Transform Structure

In this paper, a two-layer hierarchical transform structure is proposed, as shown in Fig. 1. The first transform layer uses DWT to decompose the given input block into four subbands, i.e., LL, LH, HL, and HH. The input block is a 16×16 block of MCP error. After the first transform layer, most of the residue energy is compacted to the LL subband. The second layer selects appropriate directional transform to further compact the intermediate coefficients in four subbands to four blocks in the transform domain, i.e., T_{LL}, T_{LH}, T_{HL}, and T_{HH}. Finally, the transform coefficients are quantized and entropy coded.

In the first layer, 2-D Haar transform (HT) is chosen to decompose the given 16 × 16 block into four subbands. The 1-D HT of the input signal $x(n)$, $n = 0, \ldots, 15$, can be represented by two sequences, the approximate coefficients

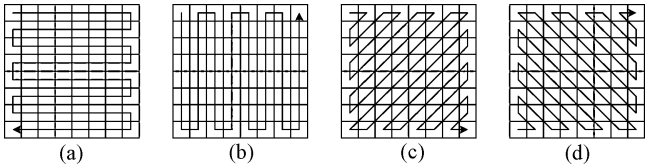


Fig. 3. Four patterns for scanning one subband in the classification step in (a) horizontal, (b) vertical, (c) 45°, and (d) 135° directions.

l(n) and detail coefficients *h(n)*, defined as follows:

$$l(n) = [x(2n) + x(2n+1)] \big/ 2, \qquad n = 0, \ldots, 7 \tag{1}$$

$$h(n) = x(2n) - x(2n+1), \qquad n = 0, \ldots, 7. \tag{2}$$

Using only integer additions, subtractions, and shifts, HT has the lowest computational cost among discrete transforms.

2-D HT is separable, so is often implemented by applying 1-D HT sequentially to the rows and columns. Given a block to be transformed, the filters for updating (1) and prediction (2) are always supported by the pixels within the block, and thus no extrapolation is needed. This merit avoids boundary effect, which is severe in other block or region-based directional DWT using longer-tap filters because of the mirror extrapolation for the supporting pixels outside the block boundary. This is the other reason of choosing HT, besides low complexity.

The motivation of the first transform layer is to quickly and efficiently compact the residue energy to the LL band. We studied the macroblocks (MBs) using the proposed directional transform, based on five 720*p* sequences, *City*, *Crew*, *Night*, *Optis*, and *ShuttleStart*, calculated the probabilities of having nonzero coefficients in the four subbands, and averaged the probabilities at different bit-rates, i.e., quantization parameter (QP) equal to 22, 27, 32, and 37. As shown in Fig. 2, the three subbands, LH, HL, and HH, have high probabilities to have all-zero blocks, whereas the LL subband has high probability to have nonzero coefficients. Similar to the coded block pattern (CBP) in H.264/AVC, four Boolean variables are used for the four subbands to indicate whether they have nonzero coefficients to be transmitted, respectively. As the distributions of the Boolean variables are far from the uniform distribution (see Fig. 2), the actual number of bits used to code one variable should be much less than 1, according to the fundamentals of information theory. Note that in H.264/AVC or the other block-based directional transforms, the four partitions in an MB, where 2-D order-8 transform is applied, have the same probability distribution of having nonzero blocks, so more bits are needed to code CBP, compared with the proposed two-layer directional transform.

B. Transform Design

After the first transform layer, the intermediate coefficients in each subband still have correlations, which are removed in the second transform layer. A set of transforms, efficiently dealing with different edge directions, are designed and applied to subbands with different edge directions.

In an MB, the energy and edge structure of the 16×16 MCP residue block are mainly preserved in the LL subband, as introduced in Section II-A. For some MBs, where the MCP along the edge is poor, the energies in the LH and HL subbands are not negligible, although very low. In this case, the edge structures in LH and HL subband are similar to that in the LL subband, according to our observation on natural videos. Therefore, the transform favoring a certain direction is applied to the four subbands in an MB. In existing block-based directional transforms, four partitions in an MB, where 2-D order-8 directional transforms are applied, may be suitable for different directions of transforms, so four signals are needed to indicate the transform directions for the four partitions, respectively. Here, the side information used to indicate the direction of the transform is reduced, as it is transmitted on an MB basis, not a block basis.

To transform the intermediate coefficients in each subband, we propose using 2-D nonseparable transform to replace the traditional separable transforms, such as 2-D DCT, because separable block-based transforms, which are performed along the horizontal and vertical directions consequently, can compact the edges along these two directional efficiently, but cannot efficiently compacts energy along other edge directions.

In a 2-D nonseparable transform, an 8×8 subband $\mathbf{S}$ is represented as a linear combination of orthonormal basis images $\boldsymbol{U_k}(0 \le k \le 63)$ that also have size 8×8, as follows:

$$\mathbf{S} = \sum_{k=0}^{63} T_k \boldsymbol{U_k} \tag{3}$$

where T_k is the transform coefficient in $\mathbf{T}$. $\mathbf{T}$, having the size 64×1, is the representation of $\mathbf{S}$ in the transform domain. The inner product of any two basis images $\boldsymbol{U}_k$ and $\boldsymbol{U}_l$ satisfies

$$\langle \boldsymbol{U}_k, \boldsymbol{U}_l \rangle = \begin{cases} 1, & \text{if } k = l \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

With an orthonormal set of basis images, one can find the transform coefficient T_k by the inner product of $\boldsymbol{U}_k$ and $\mathbf{S}$, as follows:

$$T_k = \langle \boldsymbol{U}_k, \mathbf{S} \rangle . \tag{5}$$

In the proposed directional transform, four sets of basis images, denoted as $\mathbf{U}_H$, $\mathbf{U}_V$, $\mathbf{U}_{45}$, and $\mathbf{U}_{135}$, are designed to transform the subbands that contain the dominant horizontal, vertical, 45°, and 135° edges, respectively. $\mathbf{U}_H$, $\mathbf{U}_V$, $\mathbf{U}_{45}$, and $\mathbf{U}_{135}$ are obtained as follows.

1) We choose five 720*p* (*Harbor*, *Jets*, *Raven*, *Sailormen*, *SpinCalendar*) and five 1080*p* (*BlueSky*, *PedestrianArea*, *Riverbed*, *RushHour*, *VintageCar*) high-definition (HD) sequences as the training set, which cover a wide range of content: smooth to complex textures and slow to rapid motions. The first ten frames of each sequence are coded in all-I frame mode using QP 22, 27, 32, and 37. All the intraprediction residues are collected, and transformed by the first-layer 2-D order-2 HT, in order to generate the 8×8 subbands for the next step of training. Interprediction residues are not used in the training step, because they have much lower energy and more random distribution and make the outcoming transform bases biased toward the training set.

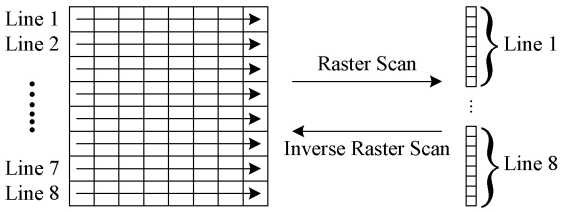


Fig. 4. Illustration of raster scan and inverse raster scan.

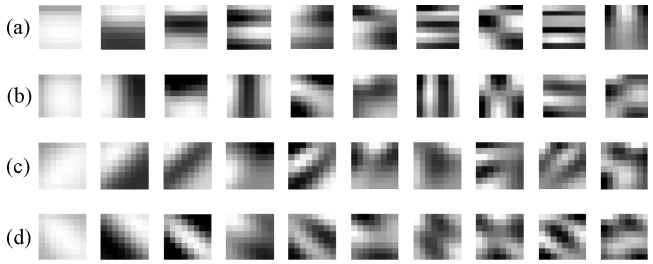


Fig. 5. First ten basis images of the transforms for four directions. (a) $\mathbf{U}_H$. (b) $\mathbf{U}_V$. (c) $\mathbf{U}_{45}$. (d) $\mathbf{U}_{135}$.

2) All the 8×8 subbands are classified into four categories, according to the dominant edge direction that a certain subband has. For classification, each 8×8 subband is scanned along the four directions as in Fig. 3 and converted to four different 64×1 vectors. The 1-D vectors are transformed using 1-D order-64 Hadamard transform, and entropy coded. The entropy coding used is the context-adaptive binary arithmetic coding (CABAC) scheme for 8×8 residual blocks in H.264/AVC. If the scan direction has higher correlation with the actual direction in the subband, the resulting 1-D vector can have less high frequency components and uses less bits. Then, the scanning direction making its 64×1 vector coded by the least number of bits is recognized as the 8×8 subband's direction.
3) The basis images for a certain category are calculated, based on the statistics of all the 8×8 subbands classified into that category. In each category, each subband $\mathbf{S}$ is raster scanned (see Fig. 4) into a 64×1 signal $\mathbf{s}$ as a sample of a random process. Then the autocorrelation matrix of the 64×1 random process is calculated as $\mathbf{R}_s$ as follows:

$$\mathbf{R}_s(m, n) = E\left[\mathbf{s}(m)\mathbf{s}(n)\right] \qquad 0 \le m, n \le 63. \tag{6}$$

Note that the size of $\mathbf{R}_s$ is 64×64. Then, $\mathbf{R}_s$'s normalized eigenvectors $\mathbf{V} = [\mathbf{v_0}, \mathbf{v}_1, \ldots, \mathbf{v}_{63}]$, of which the corresponding eigenvalues are in the descending order, are calculated, and $\mathbf{v}_n (0 \le n \le 63)$ is inverse raster scanned (see Fig. 4) into a 8×8 matrix, which is the nth basis image for the category.

As the results of the above three-step training, Fig. 5 shows the first ten basis images of the transforms for the four directions. Obviously, the first several basis images, which represent the low frequency components of blocks, have clear and gradual changes in gradient, whereas the basis images representing high frequency components become random. The transform blocks, where most of the pixels are well predicted and the prediction error along an edge becomes dominant, can be efficiently represented by linear combinations of the first a few basis images, i.e., by a few transform coefficients. Note that such blocks are common, because with the evolution of the MCP techniques the reduction of prediction error in smooth area is always more significant than that along edges.

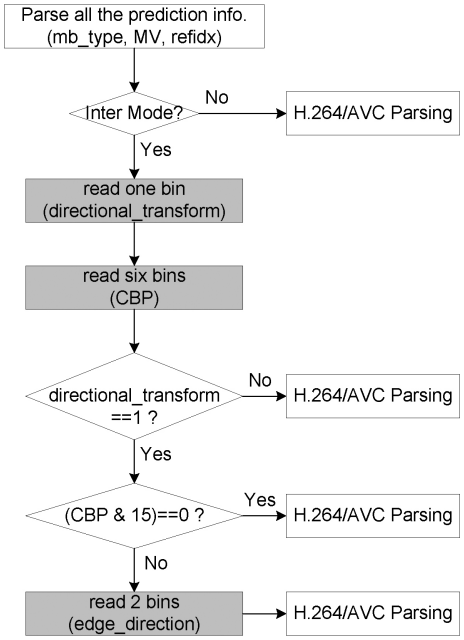


Fig. 6. Syntax modification in H.264/AVC by the directional transform.

C. Integration into H.264/AVC

The proposed directional transform is integrated into H.264/AVC as an alternative to the adaptive 2-D order-4 and order-8 ICTs therein. It can be used to transform the prediction errors by all interprediction modes, regardless of the motion partition. The numbers in the basis images, which are originally real-valued, are scaled by 2048 and rounded to the nearest integers, and the transform is implemented using 32-bit integer arithmetic. To represent one basis image, 64 integers are stored, and for a certain direction, there are 64 basis images. Therefore, the memory requirement for storing all the basis images is $64 \times 64 \times 4 = 16\,384$ integers. Since the basis images have the same norm, no scaling matrix is required. As the transform coefficients of each subband, denoted as $\mathbf{T}$, are a 1-D coefficient sequence (3), no scan is performed, and $\mathbf{T}$ is input to H.264/AVC CABAC module for entropy coding.

The decision on using the directional transform or the ICTs in H.264/AVC is based on the criterion of R-D cost. When the interprediction error for an MB is known, the four transforms favoring four directions are tried one by one, all followed by quantization, entropy coding, and reconstruction. The R-D cost is calculated by

$$C = D + \lambda R \tag{7}$$

where D is the reconstruction error of the MB and R is the number of bits used to coding the MB's residue [21]. If the

TABLE I
TEST CONDITIONS

Test Sequence	720*p* and 1080*p*
Benchmark	H.264/AVC High Profile
Coding structure	IPPP... and IBBP...
Intraframe period	Only the first frame
Entropy coding	CABAC
FME	On
R-D optimization	On
Adaptive rounding	Off
QP	I (22, 27, 32, 37) P (23, 28, 33, 38) B (24, 29, 34, 39)
Reference frame	4
Search range	±64
Frame number	58

TABLE II
R-D PERFORMANCE COMPARED WITH H.264/AVC HIGH PROFILE

HD Sequences	IPPP Coding Structure		IBBP Coding Structure	
	Δ-Bit-Rate (%)	Δ PSNR *Y* (dB)	Δ-Bit-Rate (%)	Δ PSNR *Y* (dB)
City	−4.10	0.13	−2.12	0.07
Crew	−10.29	0.29	−11.16	0.29
Night	−2.51	0.10	−2.90	0.10
Optis	−3.21	0.08	−4.17	0.10
ShuttleStart	−8.07	0.28	−8.24	0.28
Station	−5.48	0.21	−0.24	0.03
Sunflower	−11.38	0.46	−7.12	0.28
ToysCalendar	−8.78	0.21	−4.92	0.12
Tractor	−5.36	0.21	−3.68	0.14
WalkingCouple	−4.32	0.11	−3.52	0.09
Average	−6.35	0.21	−4.81	0.15

TABLE III
PROPORTIONS OF USING THE PROPOSED DIRECTIONAL TRANSFORM AT DIFFERENT BIT-RATES AND CODING STRUCTURES

	IPPP Coding Structure				IBBP Coding Structure			
QP	22	27	32	37	22	27	32	37
City	25.3	53.4	69.5	79.5	26.3	56.4	71.8	80.8
Crew	57.7	82.8	89.9	90.9	65.8	83.9	90.0	92.4
Night	26.1	39.0	46.6	57.2	27.1	41.9	53.7	60.7
Optis	51.7	68.9	76.5	80.4	28.3	67.4	84.4	86.6
ShuttleStart	56.4	69.0	73.6	76.5	58.3	68.6	74.1	80.7
Sunflower	80.3	88.6	92.4	95.6	78.6	82.9	87.4	92.2
Station	72.9	78.4	82.1	86.4	70.2	76.9	83.6	87.1
ToyCalendar	65.2	76.7	79.5	82.4	68.6	72.9	75.5	78.2
Tractor	44.0	62.9	69.0	80.4	46.9	59.3	71.9	80.4
WalkingCouple	56.1	64.3	61.7	66.4	63.6	58.5	63.0	69.7

least cost of using one of the four directional transforms is smaller than the cost of using the ICTs in H.264/AVC, the directional transform with the least-costing direction is selected. Since the encoder uses the brute-force method to find the optimal transform mode, the complexity is increased. Compared with JM16.2, the encoding time is increased by 60%.

The MB-level syntax of H.264/AVC is modified for the additional transform. As shown in Fig. 6, the shaded blocks mean where the syntax is changed. After parsing the prediction information of a certain MB, which is the same as H.264/AVC does, the parser will check whether the MB is intercoded. If it is true, the parser will read the next bin as the flag indicating whether directional transform is used. Note that bin is one binary signal output from the CABAC decoding engine, and does not necessarily correspond to one bit in the bitstream. Three context models are used to model the probability distribution of the flag "directional_transform" in three cases determined by the values of "directional_transform" in the upper and left MBs. Then, the parser will read the next six bins as CBP. If the MB uses directional transform, the less significant four bins in CBP indicate whether the four subbands, LL, LH, HL, and HH, have nonzero coefficients, respectively. In this case, four context models are used for the four bins, respectively. Otherwise, the MB does not use directional transform; the parser reads the CBP using the method specified in H.264/AVC. If the MB uses directional transform and has nonzero transform coefficients, i.e., (CBP&15) is not equal to zero, the next two bins, representing the four possible transform directions, are read. For each bin, three context models are used for the three cases determined by the values of the corresponding bins in the upper and left MBs. In summary, 13 context models are established for the additional syntax elements introduced by using directional transform.

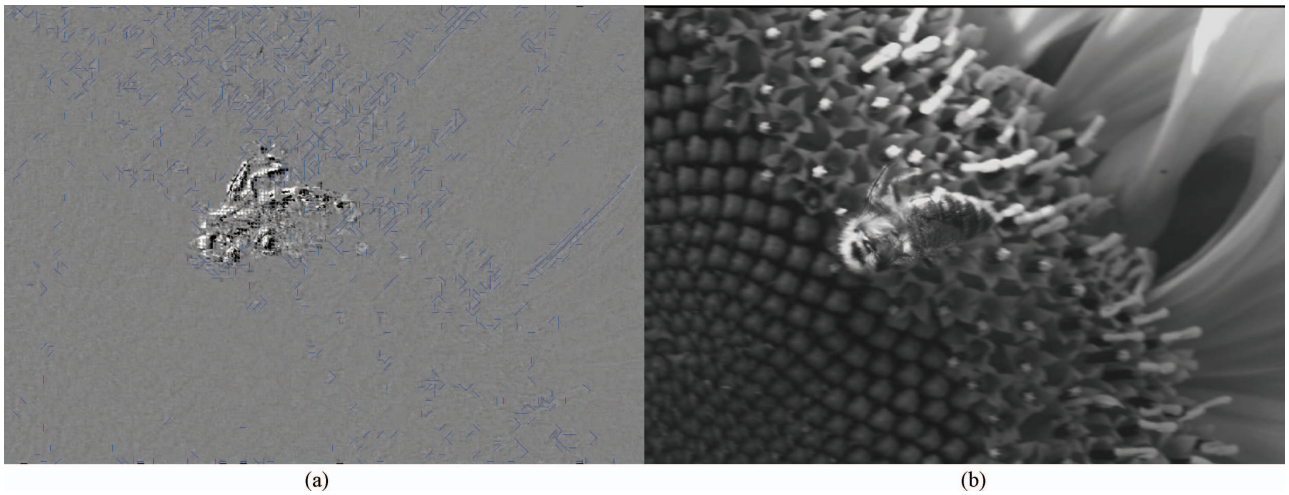
III. EXPERIMENTAL RESULTS

The proposed two-layer directional transform is integrated into H.264/AVC's reference software JM16.2, and used as an alternative to the adaptive 2-D order-4 and order-8 ICTs therein. Table I gives the test conditions; Table II shows the coding gain compared with H.264/AVC High Profile, measured by the bit-rate reduction at the same PSNR or by the PSNR gain at the same bit-rate [22]. The averages over all the test sequences are shown in the bottom row.

In the IPPP coding structure, the improvements are more than 0.2 dB on average, while for the best case of *Sunflower*, the gain is up to 0.46 dB, equivalent to 11.38% bit-rate reduction. The sequence *Sunflower* has large smooth and static areas, where MCP performs very well and the residue energy approaches zero. In the bitstream, most of the bits are used to represent small areas dominated by edges. When the directional transform is used, the prediction errors along edges, which are difficult to predict, are efficiently represented by fewer coefficients and much less bits are used to code them. As a result, the overall performance is significantly improved.

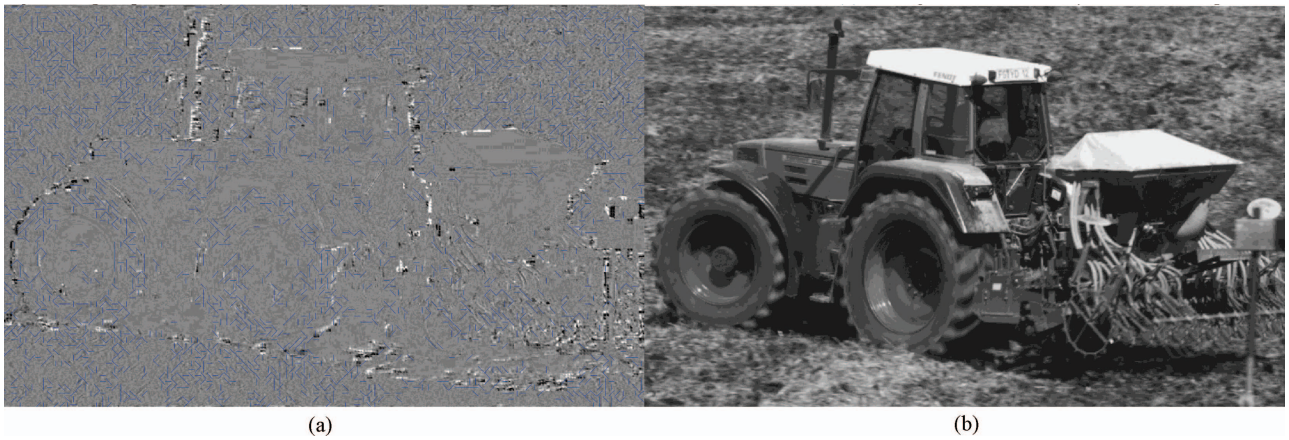
In the IBBP coding structure, which means two B-frames are used between I and P-frames, the coding efficiency is improved on an average of 0.06 dB less than that in IPPP coding structure. That is because the bitstream sizes of IBBP-coded videos are much smaller than those of IPPP-coded ones, and therefore the overhead used to indicate the transform direction, although is smaller than other directional transforms, has more negative impact on the overall performance improvement.

Table III shows the percentage of MBs coded by the proposed directional transform at four QPs. Only the MBs having nonzero luminance transform coefficients are studied. On average, more than half of the MBs are coded by the proposed directional transform, and for some sequences, such



(a) (b)

Fig. 7. 1080*p* sequence: *Sunflower*. (a) Residual frame after MCP and the directional transform used. (b) Original frame with only luminance component.



(a) (b)

Fig. 8. 1080*p* sequence: *Tractor*. (a) Residual frame after MCP and the directional transform used. (b) Original frame with only luminance component.

as *Crew* and *Station*, the percentages are more than 80%. This implies that the proposed directional transform is useful in HD video coding. At lower bit-rates, the percentages of MBs coded by the directional transform are higher than those at higher bit-rates. There are two reasons. First, at high bit-rates, the percentage of intra MBs, which never use the directional transform, is high. Second, at low bit-rates, more bits are allocated to MB headers, such as MB type, motion vectors (MVs), and ref_idx, and thus less bits are allocated to residues. In this case, the directional transform that represents edges using very few coefficients is more frequently selected. This merit of directional transform is provided by the first several basis images, as shown in Fig. 5. Note that given a sequence, the percentages of using directional transform are similar, no matter the sequence is IPPP or IBBP-coded. This phenomenon means that with both coding structures the same areas in a frame need the directional transform, but in IBBP-coded sequences, the coding gains are more canceled out by the overhead. This agrees with the observation in Table II.

Figs. 7 and 8 give two examples of how the directional transforms are used in frames. Figs. 7(b) and 8(b) are the two original frames in *Sunflower* and *Tractor*, respectively. Figs. 7(a) and 8(a) are the two residual frames after MCP, where the MBs labeled with blue lines are coded using directional transforms and the blue lines' directions indicate the transforms' directions. As can be seen, the directional transforms are frequently used in the texture areas, where the energy of the MCP error along an edge becomes dominant in a block, e.g., the grass in *Tractor*.

TABLE IV
COMPARISON OF AVERAGE DECODING EXECUTION TIME

HD Sequences	IPPP Coding Structure		IBBP Coding Structure	
	Nonzero 8 × 8 Blks (%)	ΔTime (%)	Nonzero 8 × 8 Blks (%)	ΔTime (%)
City	17.83	19.4	12.19	3.0
Crew	13.30	30.3	9.33	1.6
Night	23.37	21.0	17.37	5.6
Optis	20.54	37.8	14.08	2.5
ShuttleStart	5.79	31.6	4.37	3.5
Station	4.90	8.9	4.13	1.8
Sunflower	6.02	10.9	4.67	5.6
ToysCalendar	11.24	15.1	8.14	3.5
Tractor	24.12	24.8	19.80	7.2
WalkingCouple	24.22	9.5	17.90	7.1
Average	15.13	20.9	11.20	4.1

The proposed directional transform is nonseparable, which has higher complexity than the separable order-8 and order-4 ICTs in H.264/AVC. Given the transform size, a nonseparable transform uses four times the arithmetic operations of a separable one, if implemented by matrix multiplication. Therefore, the second layer of the directional transform has four times the complexity of the 2-D order-8 ICT in H.264/AVC. Taking the first layer 2-D HT decomposition into consideration, the complexity is even higher. We tested the execution time of decoding to show how the directional transform influences the overall complexity. The laptop used for testing has the Intel Core i5 CPU at 2.53 GHz and 2.98 GB of RAM. As shown in the third and fifth columns of Table IV, the averaged increasing execution time compared with JM16.2 decoder is 20.9% for IPPP bitstreams, and the increment is only 4% for IBBP bitstreams, because B-frames have more complicated MCP procedures, such as interpolation and MV derivation, which reduce the computation proportion of transform. The average proportion of the nonzero 8×8 luminance blocks to the total number of luminance 8×8 blocks in P and B-frames are shown in the second and fourth columns of Table IV. As can be seen, the inverse transform is applied to a relatively small proportion of blocks, but the performance gain is remarkable. At the same time, the increased execution time is acceptable.

IV. Conclusion

This paper presented a directional transform scheme for coding interprediction errors in block-based hybrid video coding. It proposed a two-layer transform structure, where the first layer uses 2-D HT to compact the residue energy to the LL band and then the second layer uses 2-D nonseparable directional transforms to deal with the arbitrary edge directions in the four subbands. By doing this, the edges were efficiently compacted to a few coefficients and the overhead used to indicate the transform direction was affordable. Experimental results showed that the proposed scheme provided PSNR gain up to 0.46 dB, compared with H.264/AVC High Profile.

References

[1] W. K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," *IEE Proc., Part I*, vol. 136, no. 4, pp. 276–282, Aug. 1989.

[2] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[3] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 2001.

[4] J. Dong, K. N. Ngan, C. K. Fong, and W. K. Cham, "2D order-16 integer transforms for HD video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 10, pp. 1463–1474, Oct. 2009.

[5] R. L. Claypoole, G. M. Davis, W. Sweldens, and R. G. Baraniuk, "Nonlinear wavelet transforms for image coding via lifting," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1449–1459, Dec. 2003.

[6] D. Taubman, "Adaptive, non-separable lifting transforms for image compression," in *Proc. IEEE ICIP*, Oct. 1999, pp. 772–776.

[7] O. N. Gerek and A. E. Cetin, "A 2D orientation-adaptive prediction filter in lifting structures for image coding," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 106–111, Jan. 2006.

[8] W. Ding, F. Wu, X. Wu, S. Li, and H. Li, "Adaptive directional lifting-based wavelet transform for image coding," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 416–427, Feb. 2007.

[9] C. Chang, A. Maleki, and B. Girod, "Adaptive wavelet transform for image compression via directional quincunx lifting," in *Proc. IEEE Workshop MMSP*, Oct. 2005, pp. 1–4.

[10] R. Eslami and H. Radha, "A new family of nonredundant transforms using hybrid wavelets and directional filter banks," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1152–1167, Apr. 2007.

[11] Y. Tanaka, M. Ikehara, and T. Q. Nguyen, "Multiresolution image representation using combined 2-D and 1-D directional filter banks," *IEEE Trans. Image Process.*, vol. 18, no. 2, pp. 269–280, Feb. 2009.

[12] Y. Liu and K. N. Ngan, "Weighted adaptive lifting-based wavelet transform for image coding," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 500–511, Apr. 2008.

[13] Y. Liu, K. N. Ngan, and F. Wu, "3-D shape-adaptive directional wavelet transform for object-based scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 888–899, Jul. 2008.

[14] B. Zeng and J. Fu, "Directional discrete cosine transforms: A new framework for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 305–313, Mar. 2008.

[15] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 1, pp. 59–62, Feb. 1995.

[16] F. Kamisli and J. S. Lim, "Transforms for the motion compensation residual," in *Proc. IEEE ICASSP*, Apr. 2009, pp. 789–792.

[17] R. Cohen, S. Klomp, A. Vetro, and H. Sun, "Direction-adaptive transform for coding prediction residuals," in *Proc. IEEE ICIP*, Sep. 2010, pp. 789–792.

[18] H. Xu, J. Xu, and F. Wu, "Lifting-based directional DCT-like transform for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1325–1335, Oct. 2007.

[19] S. Zhu, S.-K. A. Yeung, and B. Zeng, "R-D performance upper bound of transform coding for 2-D direction sources," *IEEE Signal Process. Lett.*, vol. 16, no. 10, pp. 861–864, Oct. 2009.

[20] Y. Ye and M. Karczewicz, "Improved H.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning," in *Proc. IEEE ICIP*, Oct. 2008, pp. 2116–2119.

[21] 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.

[22] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," document VCEG-M33, ITU-T SG16/Q6, Apr. 2001.

Jie Dong (S'07–M'10) received the B.E. and M.E. degrees in information engineering from Zhejiang University, Hangzhou, China, in 2002 and 2005, respectively, and the Ph.D. degree in electronic engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 2009.

In 2010, she was a Post-Doctoral Research Fellow with the Chinese University of Hong Kong. In 2011, she joined the Office of the Chief Technical Officer, InterDigital Communications, San Diego, CA, as a Staff Engineer. Her current research interests include high-efficiency video coding and real-time high-definition video processing.

King Ngi Ngan (F'00) received the Ph.D. degree in electrical engineering from Loughborough University, Loughborough, U.K.

He is currently a Chair Professor with the Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, Hong Kong. He was previously a Full Professor with the Nanyang Technological University, Singapore, and the University of Western Australia, Crawley, Australia. He holds honorary and visiting professorships of numerous universities in China, Australia, and South East Asia. He has published extensively including three authored books, six edited volumes, and over 300 refereed technical papers, and also edited nine special issues in journals. In addition, he holds ten patents in the areas of image/video coding and communications.

Prof. Ngan was an Associate Editor of the *Journal on Visual Communications and Image Representation*, and an Area Editor of the *EURASIP Journal of Signal Processing: Image Communication*. He was an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology and the *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 co-chaired the IEEE International Conference on Image Processing held in Hong Kong in September 2010. He is a fellow of IET (U.K.) and IEAust (Australia), and was an IEEE Distinguished Lecturer from 2006 to 2007.