Direction-Adaptive Partitioned Block Transform for Color Image Coding

Chuo-Ling Chang, *Member, IEEE*, Mina Makar, *Student Member, IEEE*, Sam S. Tsai, *Student Member, IEEE*, and Bernd Girod, *Fellow, IEEE*

***Abstract*—The direction-adaptive partitioned block transform (DA-PBT) is proposed to exploit the directional features in color images to improve coding performance. Depending on the directionality in an image block, the transform either selects one of the eight directional modes or falls back to the nondirectional mode equivalent to the conventional 2-D DCT. The selection of a directional mode determines the transform direction that provides directional basis functions, the block partitioning that spatially confines the high-frequency energy, the scanning order that arranges the transform coefficients into a 1-D sequence for efficient entropy coding, and the quantization matrix optimized for visual quality. The DA-PBT can be incorporated into image coding using a rate-distortion optimized framework for direction selection, and can therefore be viewed as a generalization of variable blocksize transforms with the inclusion of directional transforms and nonrectangular partitions. As a block transform, it can naturally be combined with block-based intra or inter prediction to exploit the directionality remaining in the residual. Experimental results show that the proposed DA-PBT outperforms the 2-D DCT by more than 2 dB for test images with directional features. It also greatly reduces the ringing and checkerboard artifacts typically observed around directional features in images. The DA-PBT also consistently outperforms a previously proposed directional DCT. When combined with directional prediction, gains are less than additive, as similar signal properties are exploited by the prediction and the transform. For hybrid video coding, significant gains are shown for intra coding, but not for encoding the residual after accurate motion-compensated prediction.**

***Index Terms*—Direction-adaptive entropy coding, direction-adaptive transform, discrete cosine transform, image coding, transform coding.**

I. Introduction

FOR over two decades, block transforms such as the 2-D DCT have been the key component in image coding techniques, e.g., the JPEG standard [1]. Although more recently, convolutional transforms, such as the 2-D DWT, have been proved superior [2], [3], JPEG is still the prevalent image coding format to date. Block transforms also remain an integral part in most video coding standards since they can be effectively combined with block-based motion-compensated prediction [4].

In image coding, directional features such as lines and edges have significant impact on both the objective and the subjective performance. Block transforms are typically constructed in a separable manner, by cascading a 1-D vertical transform and a 1-D horizontal transform. Separable transforms tend to do well for image detail oriented strictly horizontally and vertically, while ringing and checkerboard artifacts are likely to appear around edges of other orientations, significantly deteriorating visual quality.

To exploit directionality, the intra coding portion of the video coding standard H.264 predicts a block from previously encoded blocks using directional extrapolation [4], and the residual is then encoded via a block transform. This approach is effective, but it requires perfect synchronization of the DPCM loops of encoder and decoder. Any deviation between encoder and decoder can propagate over the image, leading to an overwhelming distortion. Therefore, it does not easily allow embedded coding, an important feature supported in the state-of-the-art image coding standard JPEG2000 [2]. Moreover, predictive coding can lead to an undesirable persistent offset or even limit-cycles [5].

The drawbacks associated with directional predictive coding can be eliminated by incorporating directional adaptation directly into the transform. To this end, several direction-adaptive transforms have been proposed based on the DWT [6]–[9], the DCT [10]–[12], and separable transforms customized from training data [13]. With a properly designed transform, the distortion is manageable and embedded coding can be achieved. Among these transforms, only the approaches in [10], [11], and [13] may be operated in a block-wise manner, i.e., each block with a size as small as 4×4, the basic unit in H.264, can be independently transformed. The DWT-based approaches in [6]–[9] require a larger support due to the longer basis functions in the DWT, and the lifting-based directional DCT in [12] extends across block boundaries to realize direction adaptation.

In this paper, we present the direction-adaptive partitioned block transform (DA-PBT), based on our previous work in [14]. It can be viewed as a direction-adaptive block transform, constructed similarly as the directional DCT in [10] and [11], with additional direction-adaptive block partitioning to improve energy concentration while reducing complexity, and more efficient coefficient ordering for entropy coding. For image coding, the proposed DA-PBT significantly outperforms the directional DCT while demanding less computation [10], [11]. As a block transform, the DA-PBT can be directly incorporated into the prediction-based video coding standards to work with the block-

Manuscript received November 03, 2008; revised September 18, 2009. First published March 08, 2010; current version published June 16, 2010. This work was supported by the Max Planck Center for Visual Computing and Communication. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Oscar C. Au.

The authors are with the Information Systems Laboratory, Stanford University, Stanford, CA, 94305, USA (e-mail: chuoling@stanford.edu).

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/TIP.2010.2044964

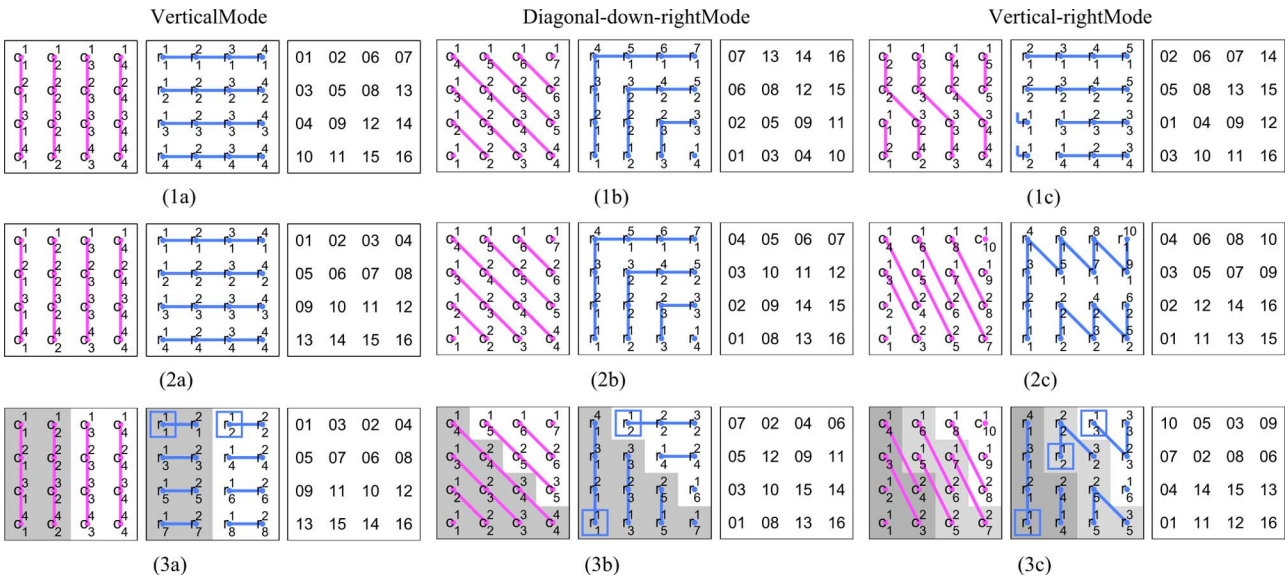

Fig. 1. Transform directions and the encoding order of the DDCT, the DA-BT and the DA-PBT. In (1c), r_1^1 and r_2^1 are also connected to r_1^2 and r_2^2, respectively, but the lines are omitted for clarity. (1a) DDCT, (1b) DDCT, (1c) DDCT, (2a) DA-BT, (2b) DA-BT, (2c) DA-BT, (3a) DA-PBT, (3b) DA-PBT, (3c) DA-PBT.

based intra prediction as well as the block-based motion-compensated interframe prediction [4]. Due to limited space, a theoretical analysis of the proposed direction-adaptive transform is not included here, but can be found in [15].

In the remainder of this paper, we briefly review related work in Section II. The proposed DA-PBT is then presented in Section III. The application of the DA-PBT to color image coding is described in IV, followed by a discussion of combining the DA-PBT with prediction-based image and video coding in Section V. Experimental results are shown in Section VI.

II. Related Work

In conventional image coding, a 2-D DCT is composed of two stages of separable 1-D transforms. Taking the 4×4 DCT as an example, the four columns in the 4×4 block shown in Fig. 1(1a) are first transformed in Stage 1 into columns of DCT coefficients, $\{c_i^{1\sim4}\}$, $i = 1, \ldots, 4$. Subsequently, in Stage 2 the rows of coefficients are further transformed into $\{r_j^{1\sim4}\}$, $j = 1, \ldots, 4$. To encode the DCT coefficients, a zigzag scanning order is adopted, as indicated in the rightmost plot in Fig. 1(1a), so that the coefficients generally decrease in magnitude along the scan. This property can then be exploited in entropy coding of the quantization indices [1], [16], as we shall discuss in Section IV-B.

To our knowledge, the work most closely related to our proposed DA-PBT is the directional DCT (DDCT) [10], [11]. The DDCT consists of directional modes defined similarly as the intraprediction modes of H.264 [4], in addition to the conventional 2-D DCT. The nondirectional mode, the vertical and the horizontal mode in the DDCT all correspond to the conventional transform [Fig. 1(1a)], whereas the other modes, however, are constructed by varying the transform directions. For instance, as shown in Fig. 1(1b), in Stage 1 of the diagonal-down-right mode, the 1-D DCTs, possibly with different lengths, are applied to the sequences along the corresponding direction. In Stage 2, another set of the 1-D DCTs is further applied to the set of coefficients from Stage 1 with the same superscript index, e.g., $\{c_{1\sim7}^1\}$, because they represent the frequency components at similar spectral locations and are likely to exhibit higher correlation [11]. To encode the resulting coefficients, a zigzag scanning order modified from that in Fig. 1(1a) is adopted, as shown in Fig. 1(1b). Similarly, the transforms and the scanning order in the vertical-right mode of the DDCT are illustrated in Fig. 1(1c). Note that the Stage-1 direction in the DDCT does not match exactly with that defined in the vertical-right mode of H.264 , i.e., $\triangle x/\triangle y = 1/2$. The remaining modes can be derived by flipping the operations in the diagonal-down-right or the vertical-right mode in the appropriate dimensions.

III. Direction-Adaptive Partitioned Block Transform (DA-PBT)

A. Direction-Adaptive Block Transform

To efficiently represent blocks containing directional image features, the proposed DA-PBT combines a direction-adaptive block transform (DA-BT) similar to the DDCT with a direction-adaptive block partitioning scheme, which shall be discussed in Section III-B. There are two key differences between the proposed DA-BT and the DDCT in [11].

The first difference is in the choice of the transform directions in the vertical-right mode (and the three associated modes with flipped directions). We argue that the direction in Stage 1 of the DDCT [Fig. 1(1c)] is still close to the vertical direction, and therefore the vertical-right mode may not render much benefit over the nondirectional or the vertical mode. To provide a directional selectivity that covers all possible feature orientations more evenly, we adopt the configuration in Fig. 1(2c). Aside from the above consideration, the adopted direction matches that in H.264 [4], allowing a simpler implementation when the DA-BT is combined with intra prediction, as we shall discuss in Section V-A.

The second difference is not in the transform itself, but in the ordering when encoding the coefficients. Consider the diagonal-down-right mode of the DDCT shown in Fig. 1(1b). In general, this mode is selected if edges in the corresponding orientation appear in the block. Since the 1-D transforms in Stage 1 are aligned with the edges, energy is concentrated towards the DC coefficients, i.e., $\{c_{1\sim7}^1\}$, and the sequence containing the DC coefficients approximately consists of one or multiple step transitions. Consequently, after Stage 2, most energy still resides in $\{r_1^{1\sim7}\}$ and the low-frequency components are more likely to have larger magnitude due to the spectral characteristics of the step function. Therefore, different from the modified zigzag order used in the DDCT [Fig. 1(1b)], based on this observation we propose the encoding order shown in Fig. 1(2b). With the same argument, instead of applying the same operations for the vertical (or horizontal) mode and the nondirectional mode as in the DDCT, the encoding order of the vertical mode is modified as shown in Fig. 1(2a), although the transform remains the same. Note that these heuristic encoding orders have been further justified both empirically as will be shown in Section VI-A and theoretically in terms of their linear approximation performance using a statistical image model [15].

In addition to the nondirectional mode that corresponds to the 2-D DCT, the DA-BT consists of eight directional modes: vertical, vertical-right, diagonal-down-right, horizontal-down, horizontal, horizontal-up, diagonal-down-left, and vertical-left, all can be derived from the configurations in Fig. 1. The configurations for 4×4 blocks can be directly extended to 8×8 or 16×16 blocks. To illustrate the performance of the DA-BT, an 8×8 block containing diagonal edges is used as an example and the coefficients resulting from the conventional 2-D DCT and the DA-BT are shown in Fig. 2(a) and (b), respectively. In this example, many of the 2-D DCT coefficients still retain large magnitude. Additionally, the coefficient magnitude does not generally decrease along the zigzag order, making entropy coding designed based on this assumption inefficient. On the contrary, the DA-BT concentrates the energy to the coefficients located at the first column and the first row while keeping the others zero. It is also evident that the encoding order proposed in Fig. 1(2b) better exploits the distribution of the coefficients than the order adopted in DDCT [Fig. 1(1b)].

Note that in the 2-D DCT, the DC level of a block only affects the DC coefficient, i.e., r_1^1 in Fig. 1(1a). However, for the DA-BT as well as the DDCT, the DC level may contribute to other coefficients, e.g., $\{r_1^{2\sim7}\}$ in Fig. 1(1b), due to the unequal lengths of the Stage-1 transforms as discussed in the shape-adaptive DCT literature [17]. This leakage of the DC energy into the nonDC coefficients hampers energy concentration. To eliminate the problem, the DC separation procedure proposed in [17] is adopted in both the DDCT and the DA-BT . To transform a block, the block mean, denoted by m, is first subtracted from all pixels. After two stages of the transform, the DC coefficient r_1^1 is then set to $\sqrt{N}m$ where N denotes the number of pixels in the block, equal to the DC coefficient of the 2-D DCT. It has been shown in [17] that the resulting transform is reversible using an additional correction procedure. With DC separation/correction, the DC level affects only the DC coefficient, and, in particular, a constant block leads to at most one nonzero coefficient.

B. Direction-Adaptive Block Partitioning

Compared to the 2-D DCT, one disadvantage of both the DDCT and the DA-BT is the increase in the maximum length of the constituent 1-D DCTs. For an $S \times S$ block $(S > 2)$, the 2-D DCT requires length-S 1-D DCTs, whereas the DDCT and the DA-BT require up to length-$(2S-1)$ [Fig. 1(1b)] and length-$(3S-2)$ DCTs [Fig. 1(2c)], respectively. The increase in complexity can be significant for a large blocksize, e.g., $S = 16$. To reduce the complexity of the DA-BT, we propose to further divide the block into directional partitions, indicated by different shades in Fig. 1(3b) and (3c). The partition boundaries are aligned with the transform direction in Stage 1, and we limit the Stage-2 transforms so that they do not extend across partition boundaries. In fact, with the proposed partitioning that divides a block into 2 partitions for the diagonal-down-right mode Figs. 1(3b) and 3 partitions for the vertical-right mode [Fig. 1(3c)], the maximum required length is S, equal to that of the 2-D DCT. To further exploit the correlation among partitions, a Stage-3 transform is applied to the DC coefficients resulting from Stage 2, indicated by the squares in Fig. 1-(3). Additionally, the encoding order is modified to cope with the partitioning. The sequences that originally belong to the same sequence in the un-partitioned DA-BT are encoded in an interleaved manner. For instance, in Fig. 1(3b), $\{r_3^{1\sim3}\}$ and $\{r_4^{1\sim2}\}$ are ordered as r_3^1, r_4^1, r_3^2, r_4^2, and r_3^3. We refer to the partitioned version of the DA-BT as the DA-PBT.

To accommodate the addition of the partitioning and the Stage-3 transform, we propose a two-level DC separation/correction procedure similar to the one-level procedure in [17]. To apply the transform, the block mean, μ, is first subtracted, and the mean in each block-mean-removed partition, denoted by μ_p, $p = 1 \cdots P$ where P is the number of partitions, is further removed, followed by the Stage-1 and Stage-2 transforms. The resulting Stage-2 DC coefficients, e.g., r_1^1 and r_2^1 in Fig. 1(3b), are then set to $\sqrt{N_p}\mu_p$ where N_p is the number of pixels in the corresponding partition. Finally, the Stage-3 transform is applied and the resulting DC coefficient is replaced by $\sqrt{N}\mu$ where N is the number of pixels in the block.

To reconstruct the block, the DC coefficient is first replaced by 0 followed by the inverse Stage-3 transform. A DC correction procedure similar to that in [17] is then applied to correct the resulting Stage-2 DC coefficients. After performing the inverse Stage-2 transforms in each partition, the Stage-1 DC coefficients, e.g., $\{c_{1\sim7}^1\}$ in Fig. 1(3b), are then corrected, followed by the inverse Stage-1 transforms. This DC separation/correction procedure ensures the reversibility of the transform. Furthermore, a constant block results in at most one nonzero coefficient after three stages of the transform, and a constant partition also leads to at most one nonzero coefficient in the partition, an additional property achieved through the two-level procedure. Note that the procedure causes a slight deviation from the orthonormality of the transform. Quantitatively, for unit-variance white noise in the transform coefficients, the consequent noise variance in the reconstructed block is below 1.008, 1.003, and 1.001 for 4×4, 8×8, and 16×16 blocks, respectively, for all the modes in the DA-PBT.

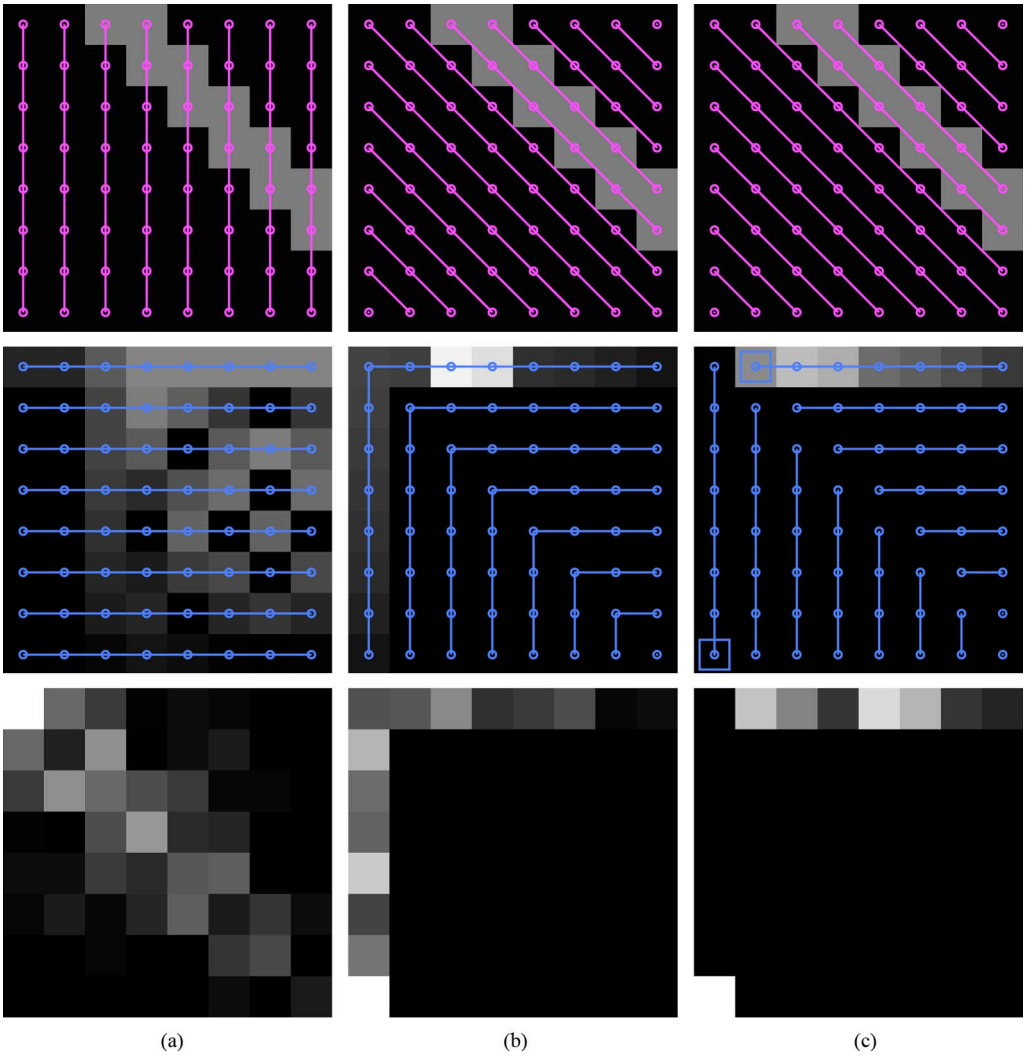

Fig. 2. Transform coefficient magnitude of an 8×8 image block using the diagonal-down-right mode if applicable. The original block is shown at the top, together with the Stage-1 directions. The magnitude of the coefficients resulting from Stage 1 is shown in the middle, together with the Stage-2 directions, and the magnitude of the final coefficients is shown at the bottom. In (c), the coefficients involved in Stage 3 of the DA-PBT are indicated by the squares. (a) 2-D DCT, (b) DA-BT, and (c) DA-PBT.

The proposed partitioning not only reduces the complexity, but also improves energy concentration. For instance, consider again the image block in Fig. 2 where the edges are contained only in the upper-right partition of the diagonal-down-right mode in the DA-PBT . The energy of the DA-PBT coefficients shown in Fig. 2(c) is mostly confined in the upper-right partition whereas the other partition consists of only one nonzero coefficient, exhibiting improved concentration compared to the DA-BT and the 2-D DCT. A similar observation can be made with the image block and the corresponding coefficients in Fig. 3. Moreover, the partitioning also improves visual quality of lossy reconstructions because the compression artifacts generally appear only in the partition containing the edges rather than the whole block. To further improve energy concentration and visual quality for vertical and horizontal image features, the partitioning can be extended to the vertical and the horizontal mode by dividing the block into two rectangular partitions as illustrated in Fig. 1(3a). To provide an example, assuming that a vertical-ish image feature appears in the left half of a 4×4 block and the right half remains constant. In the vertical mode of the DA-PBT illustrated in Fig. 1(3a), after the Stage-1 and Stage-2 transforms, because the right partition never interacts with the image feature, the energy of the feature is still confined in the left partition, affecting up to eight coefficients, whereas in the right partition only the DC coefficient r_2^1 is nonzero.

The basis functions of the 8×8 DA-PBT for different modes are shown in Fig. 4, together with the magnitude of the corresponding frequency responses. In the directional modes, it is clear that the bases are divided into multiple sets, each responsible for a partition in the block. As discussed, this helps to confine the energy of image features spatially within the associated set of coefficients. Additionally, the directional frequency bands resulting from the directional bases also help to confine the energy of directional features spectrally in a few coefficients.

C. *Quantization*

Given the quantization stepsize Q that controls the quality of the reconstruction, a transform coefficient c is mapped to the

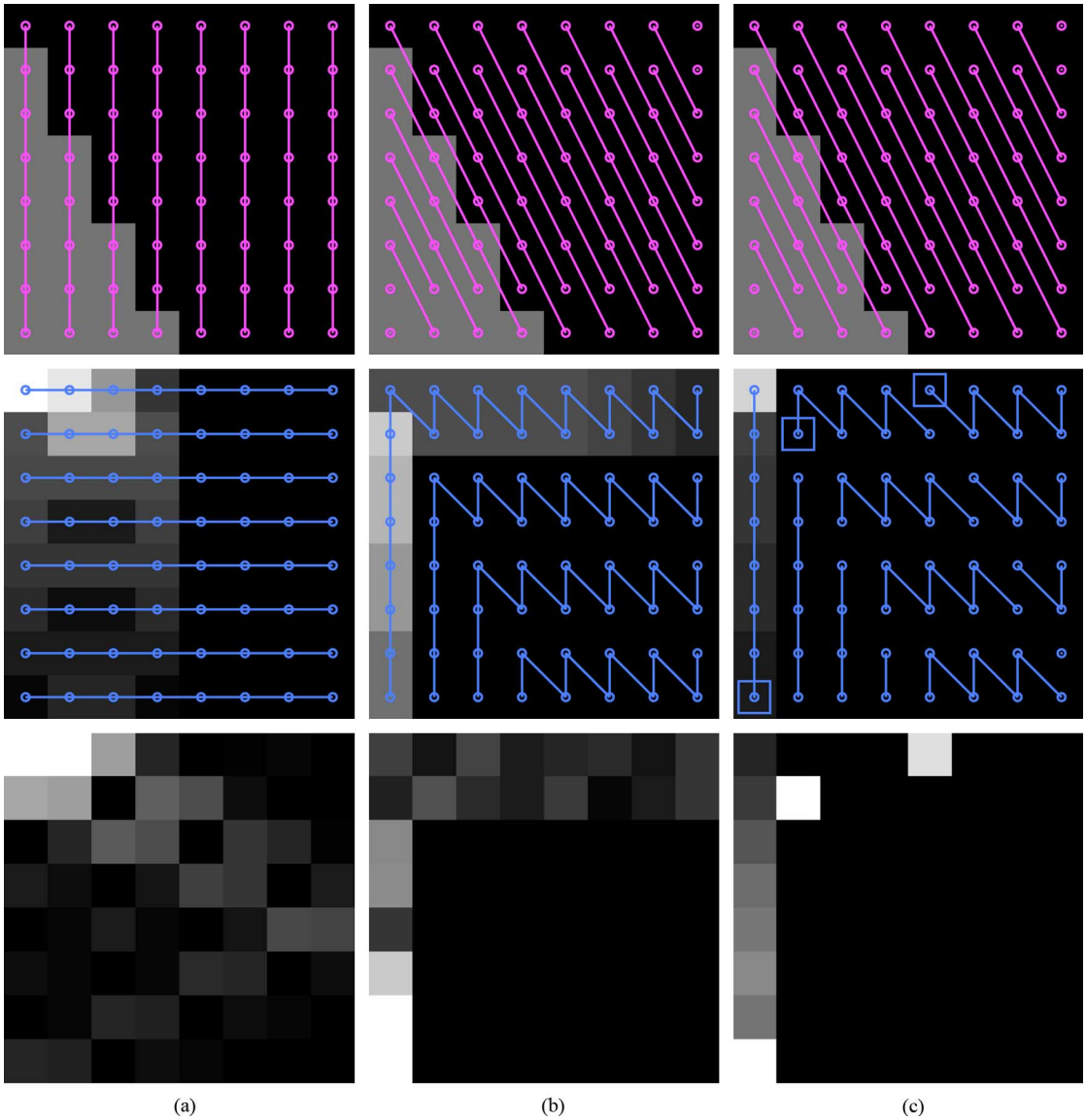

Fig. 3. Transform coefficient magnitude of an 8×8 image block using the vertical-right mode if applicable. The original block is shown at the top, together with the Stage-1 directions. The coefficients resulting from Stage 1 are shown in the middle, together with the Stage-2 directions, and the final coefficients are shown at the bottom. In (c), the coefficients involved in Stage 3 of the DA-PBT are indicated by the squares. (a) 2-D DCT, (b) DA-BT, and (c) DA-PBT.

quantization index η by $\eta = \lfloor c/Q + \Delta \rfloor$, and then reconstructed to the representative level $\hat{c} = \eta Q$. In JPEG, $\Delta = 1/2$ so that every quantization interval has the same size and the representative levels are always located in the middle of a quantization interval [1]. To better suit the skewed distribution of the transform coefficients, we use $\Delta = 1/3$ so that the interval containing zero is larger, and the representative levels of the nonzero intervals are closer to the boundary with the lower value, similar to the H.264 reference software [18].

In general, the same quantization stepsize Q is used for all coefficients to optimize the mean-squared-error rate-distortion performance. However, human visual perception tends to be less sensitive to the amplitude change in the high-frequency patterns, such as the 2-D DCT basis functions corresponding to the high-frequency coefficients shown in Fig. 4(a) [19], [20]. Consequently, for a fixed-rate budget, these coefficients can bear more quantization noise than the others. To take advantage of this property, JPEG has suggested 8×8 quantization matrices, obtained via subjective evaluations as illustrated in the rightmost plot of Fig. 4(a) for the luminance component, that can be scaled to determine the quantization stepsizes for different coefficients [1].

To design the quantization matrices for the eight directional modes of the DA-PBT, we propose the following approach that avoids cumbersome subjective tests typically required in other approaches. Assume that a quantization matrix for the 8×8 2-D DCT, where the 64 entries in the matrix are denoted by $\mathbf{q}_c \in \mathbb{R}^{64}$, is already available and achieves the best visual quality in the reconstructed block. The quantization matrix suggested in the JPEG standard [1] and shown in Fig. 4(a) could be used for that purpose. Our goal is to choose the quantization matrix for a directional transform, represented by $\mathbf{q}_d \in \mathbb{R}^{64}$, such that the covariance matrix of the quantization noise in the reconstructed block is close to that resulting from the conven-

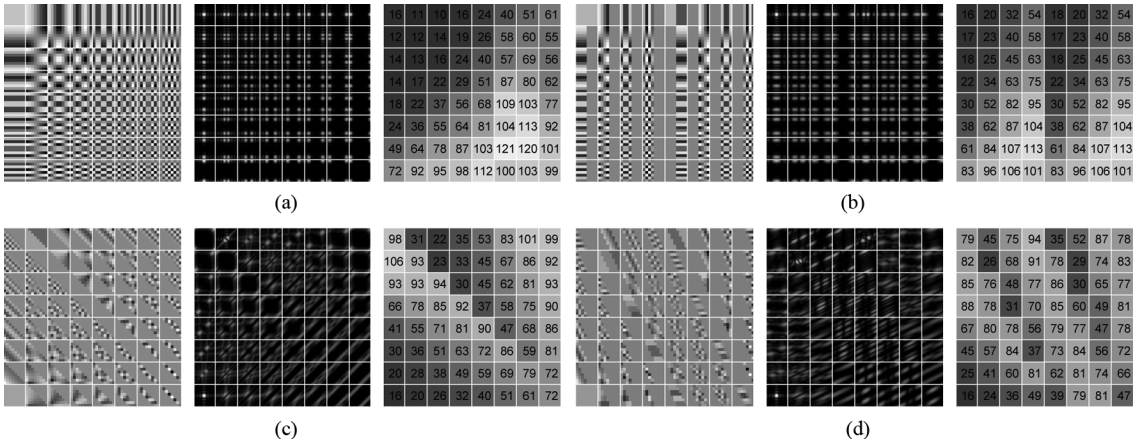

Fig. 4. For each mode of the 8×8 DA-PBT , the plots from left to right show the basis functions and the magnitude of the corresponding frequency responses of the forward transform, and the quantization matrix (luminance). (a) Nondirectional mode (2-D DCT), (b) vertical mode, (c) diagonal-down-right mode, and (d) vertical-right mode.

tional transform. Let the random variable $\mathbf{x} \in \mathbb{R}^{64}$ represent the pixels in a block, and $\mathbf{T}_c$ and $\mathbf{T}_d \in \mathbb{R}^{64\times 64}$ denote the transform matrix of the conventional and the directional transform, respectively, such that $\mathbf{T}_c\mathbf{x}$ and $\mathbf{T}_d\mathbf{x}$ represent the corresponding transform coefficients. Additionally, denote the quantization noise in $\mathbf{T}_c\mathbf{x}$ and $\mathbf{T}_d\mathbf{x}$ by $\mathbf{n}_c$ and $\mathbf{n}_d$, respectively.

To simplify the problem, we assume fine quantization such that $E[\mathbf{n}_c\mathbf{n}_c^T]$ and $E[\mathbf{n}_d\mathbf{n}_d^T]$ are diagonal, i.e., the transform-domain quantization noise is uncorrelated, and the diagonal entries are proportional to the squared value of the corresponding quantization stepsizes. To meet the aforementioned objective, we minimize the sum of squared differences between the entries in the two covariance matrices of the noise in the reconstruction, i.e., $\mathbf{T}_c^{-1}E[\mathbf{n}_c\mathbf{n}_c^T]\mathbf{T}_c^{-T}$ and $\mathbf{T}_d^{-1}E[\mathbf{n}_d\mathbf{n}_d^T]\mathbf{T}_d^{-T}$. Denote $D_2(\mathbf{q}_c)$ and $D_2(\mathbf{q}_d)$ as the diagonal matrices where each diagonal entry is the squared value of the corresponding element in $\mathbf{q}_0$ and $\mathbf{q}_d$, respectively, and

$$\mathbf{C}_c = \mathbf{T}_c^{-1}D_2(\mathbf{q}_c)\mathbf{T}_c^{-T}, \quad \mathbf{C}_d = \mathbf{T}_d^{-1}D_2(\mathbf{q}_d)\mathbf{T}_d^{-T}. \tag{1}$$

The problem can then be formulated as

$$\arg\min_{\mathbf{q}_d} \; Tr((\mathbf{C}_d - \mathbf{C}_c)(\mathbf{C}_d - \mathbf{C}_c)^T). \tag{2}$$

Since $\mathbf{T}_d$ is very close to orthonormal as discussed in Section III-B, $\mathbf{T}_d\mathbf{T}_d^T \approx \mathbf{I}$ and, therefore, see (3), shown at the bottom of the page. From (3), the solution of the the original objective (2) can be closely approximated by the square root of the diagonal entries in $\mathbf{T}_d\mathbf{C}_c\mathbf{T}_d^T$. The rounded luminance quantization matrices for the directional modes in the 8×8 DA-PBT derived from the JPEG luminance quantization matrix using this method are shown in Fig. 4(b)–4(d). The chrominance matrices can be similarly derived from the matrix suggested in JPEG [1]. For the 4×4 and 16×16 transforms, the 8×8 matrix in JPEG is first downsampled or upsampled to generate the quantization matrices for the conventional transform, and those for the directional transforms are then derived using the same method.

IV. Image Coding With DA-PBT

A. Direction Selection

To encode an image, we divide the image into 16×16 macroblocks. Each macroblock may contain a single 16×16 block or four 8×8 blocks, and each 8×8 block can be further divided into 4×4 blocks. Every block is assigned with one of the nine modes of the direction-adaptive transform with the same size as the block. When blocksize 8×8 or 4×4 is selected, a 2×2 or 4×4 2-D DCT is applied to the DC coefficients in the constituent blocks so that only one overall DC coefficient remains in a macroblock. Additionally, to exploit the correlation across macroblocks, an extra 2-D DCT is applied to the DC coefficients of every 4×4 macroblocks.

An integer quantization parameter $Q_{H.264}$ from 0 to 51 that can be directly mapped to a quantization stepsize Q as defined in H.264 is first determined to set the desired reconstruction quality [4]. Given $Q_{H.264}$, for each macroblock the blocksize and the modes are selected by minimizing a Lagrangian cost function $D_c + \lambda(R_c + R_s)$, similar to the rate-distortion optimized framework for motion estimation in video coding [21]. In the cost function, D_c denotes the distortion (sum of squared error) in the reconstructed macroblock, R_c and R_s denote the number of bits required to encode the quantization indices and the overhead signaling the selection, respectively, and λ is the

$$\begin{aligned} Tr((\mathbf{C}_d - \mathbf{C}_c)(\mathbf{C}_d - \mathbf{C}_c)^T) \approx & Tr(\mathbf{T}_d(\mathbf{C}_d - \mathbf{C}_c)\mathbf{T}_d^T\mathbf{T}_d(\mathbf{C}_d - \mathbf{C}_c)^T\mathbf{T}_d^T) \\ = & Tr((D_2(\mathbf{q}_d) - \mathbf{T}_d\mathbf{C}_c\mathbf{T}_d^T)(D_2(\mathbf{q}_d) - \mathbf{T}_d\mathbf{C}_c\mathbf{T}_d^T)^T) \end{aligned} \tag{3}$$

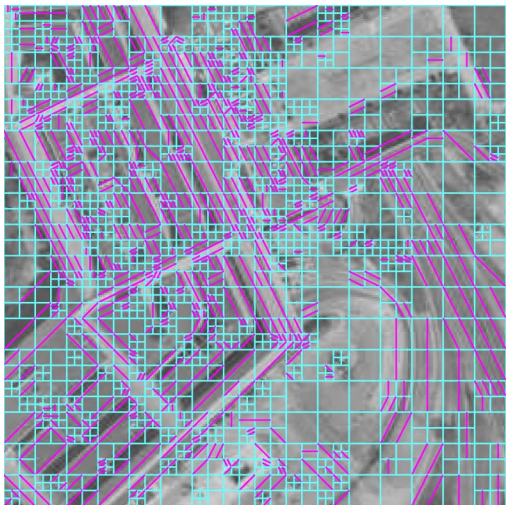

Fig. 5. Selected blocksizes and modes for a 256 × 256 region of the *Pentagon* image. The blocksizes are indicated by the square blocks, and the directional modes are indicated by the additional lines delineating the direction-adaptive block partitions. A square block without further partitions represents the nondirectional mode. The overhead signaling these selections is coded at 0.04 bpp, around 5% of the total rate.

Lagrangian multiplier set to $0.85 \cdot 2^{(Q_{H.264}-12)/3}$ obtained empirically in the context of hybrid video coding [21], [22].

An example of the selected blocksizes and modes is shown in Fig. 5. In our experiments, the overhead signaling the blocksizes and the transform modes typically takes about 5% of the total rate for most images. The figure also shows that the DA-PBT can be viewed as a generalization of the variable block-size transforms in [23] with the inclusion of nonrectangular partitions and directional transforms. As a postprocessing step, the blocking artifacts typically observed in block-transform-based image coding are mitigated by an adaptive deblocking filter modified from that proposed for the 4×4 blocks in H.264 to accommodate the 8×8 and 16×16 blocks [24].

To apply the DA-PBT to color images, the image is transformed to the YC_bC_r color space. As the human visual system is less sensitive to fine color details, the C_b and C_r channels are downsampled by a factor of 2 in both directions. By first selecting directions individually for each channel, we observed that the resulting direction representations in the two chrominance channels are highly correlated due to the shared resolution and low variation in the intensity, whereas the luminance usually contains more distinct features and requires a more detailed representation. Therefore, joint direction selection is adopted for C_b and C_r by including the rate and distortion for both channels in the cost function $D + \lambda R$. This joint selection results in a common transform mode for both chrominance signals and the overhead is reduced by 3% to 5% compared to individual selection while incurring negligible loss in the reconstruction quality.

B. Entropy Coding

To encode the quantization indices, context-based adaptive binary arithmetic coding (CABAC) used in H.264 for 4×4 blocks is adopted and further extended to handle 8×8 and 16×16 blocks [16]. Together with the encoding order illustrated in Fig. 1, CABAC exploits the common patterns of trailing ones and zeros along the ordered indices to improve compression.

The overhead signaling the blocksizes and the modes is also encoded using CABAC, similar to encoding the motion information in H.264 [16]. Specifically, one symbol for each macroblock indicates if the macroblock is divided into four 8×8 blocks, and another symbol for each 8×8 block indicates if it is further divided. To encode the mode of a block, one symbol first signals if the selected mode is the nondirectional mode. If this is not the case, we first represent each directional mode by a number from 1 for the vertical mode counterclockwise to 8 for the vertical-left mode, and also represent the nondirectional mode by 0. Additionally, denote the modes previously selected for the left, the top, the top-left, and the top-right neighbor of the current block, i.e., the blocks in the causal neighborhood, by m_l, m_t, m_{tl} and m_{tr}, respectively. If any of these neighbors exists and uses a directional mode, the mode of the current block is predicted from the neighbors, in modulo-8 arithmetic, and only the prediction residual is signaled. Otherwise the mode is signaled directly without prediction. To come up with the prediction, denoted by m_p, we use the following rules. If the left neighbor exists and it has a mode corresponding to a horizontal-ish direction, i.e., $4 \leq m_l \leq 6$, we select $m_p = m_l$ because it is likely that a horizontal-ish image feature extends from the left neighbor to the current block. If the condition for m_l does not hold, similar conditions are evaluated for m_t, m_{tl} and m_{tr} in order. If none of the above conditions holds, we continue with the following rules. If none of m_l, m_t and m_{tl} is 0, we select $m_p = m_l$ if m_{tl} is closer to m_t than to m_l, again in modulo-8 arithmetic, i.e., the modes are closer between horizontal neighbors than vertical neighbors, and $m_p = m_t$ otherwise. Instead, if any of m_l, m_t and m_{tl} is 0, $m_p = m_l$ if $m_l \neq 0$, $m_p = m_t$ if $m_l = 0$ and $m_t \neq 0$, and otherwise the current mode is directly signaled without prediction.

C. Computational Complexity

The rate-distortion optimized direction selection process generally requires performing the transform, quantization, and entropy coding for every mode to evaluate the rate and the distortion. Therefore, the computational complexity of image coding with DA-PBT at the encoder is approximately nine times that of conventional image coding using the 2-D DCT. Note that the decoder complexity is about the same as a conventional decoder since only the mode selected at the encoder is performed at the decoder.

To reduce the encoder complexity, one approach is to replace the rate and distortion evaluation with a simpler measurement such as the sum of absolute values (SAV) of the transform coefficients. The original Lagrangian cost $D_c+\lambda(R_c+R_s)$ becomes $S_c + \sqrt{\lambda}R_s$ where S_c denotes the SAV, a simplification often used in video coding implementation [21], [22]. This simplification bypasses the need for quantization and entropy coding of the coefficients for every mode, however, in our experiments, it can lead to significant loss in performance.

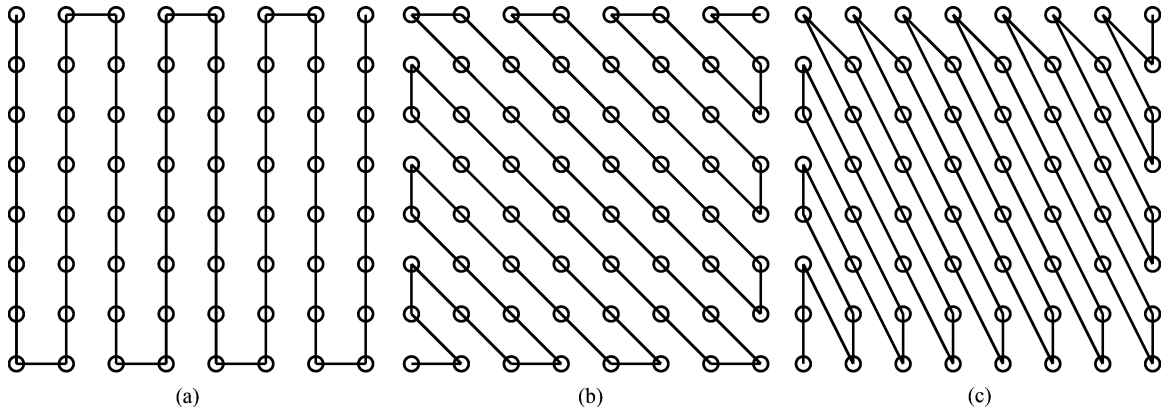

Fig. 6. Directional zigzag scanning order to estimate the gradient along the (a) vertical, (b) diagonal-down-right, and (c) vertical-right direction.

Another approach is to examine only a subset of the modes. In particular, we include in the selection process only the nondirectional mode and at most one directional mode obtained as follows. For each directional mode, a directional zigzag scanning order is defined as illustrated in Fig. 6 using 8×8 blocks as an example, and the directional gradient is measured as the sum of absolute difference between every two consecutive image pixels along the order, calculated using simple integer operations. The direction with the smallest gradient is referred to as the estimated direction, and the corresponding mode is included in the selection process if the following two conditions hold. First, the largest gradient exceeds a threshold, e.g., four times the number of pixels in the block, so that the block indeed contains significant variations that may be better handled by a directional transform. Second, at least one of the directions with the second or the third smallest gradient is an immediate neighbor to the estimated direction. This condition suggests that the block contains image features with a single orientation, e.g., lines and edges, close to the estimated direction, rather than complex patterns having multiple dominant orientations, e.g., corners, that cannot be exploited efficiently by the directional transform. By considering at most one additional mode, this approach significantly reduces the encoder complexity to less than twice of the complexity of a conventional encoder while incurring only a small performance loss, as will be shown in Section VI.

V. Residual Image Coding With DA-PBT

Hybrid coding is a technique widely used in image and video coding [4]. It consists of two steps. In the first step, the block to be encoded is predicted by a prediction block generated from the reconstruction of the previously encoded pixels. The reconstruction requires a decoder loop to be included at the encoder. Hence, this step is referred to as closed-loop prediction. Depending on the source of these pixels, there are two types of closed-loop prediction: intra prediction, using pixels in the same image, and inter prediction, using pixels in other images, e.g., previously encoded video frames. In the second step of hybrid coding, the residual block is further decorrelated by transform coding, where block transforms are especially suitable because of the block-based prediction. Assuming directional features exist in the block to be encoded, for both intra and inter prediction, directionality may still remain in the residual block due to limited prediction accuracy. This directionality can be exploited by the DA-PBT to improve the overall performance of hybrid coding.

A. Intra Prediction

Directional intra prediction is included in H.264. To encode the current block, a prediction block is extrapolated along a certain direction from previously encoded pixels. Eight directional modes are defined in H.264 for 4×4 blocks, aligned exactly to the Stage-1 directions in the DA-PBT, together with the nondirectional mode (DC mode) where the prediction is simply the average of the surrounding pixels [4]. We argue that the direction selected for intra prediction is typically close to the directionality in the original block, and therefore also coincides the remaining directionality in the residual. Based on this consideration, instead of applying a conventional nondirectional transform to the residual, we propose to always apply the DA-PBT using the same mode as intra prediction. This method requires no additional search for the transform mode and, thus, no extra signaling is needed and the computational complexity is approximately the same as if the 2-D DCT is always applied. Moreover, since the DC energy is typically small in the residual, DC separation/correction and the additional 2×2 or 4×4 2-D DCT applied to the DC coefficients within a microblock (Section IV-A) can be omitted. To further reduce the complexity, directional intra prediction and the DA-PBT along the same direction can be combined by applying the DA-PBT first and then performing prediction only to the Stage-1 DC coefficients (with proper scaling), rather than to all pixels in the block. Note that we extend the nine intra prediction modes for 4×4 blocks defined in H.264 to 8×8 and 16×16 blocks to be applied in conjunction with the DA-PBT of the corresponding blocksize.

B. Inter Prediction

In most video coding standards, inter prediction is achieved through block-based motion-compensated prediction [4]. Analogous to the argument for intra prediction, if directional features exist in the block to be encoded, similar directionality generally resides in the corresponding prediction block, generated from the previously encoded video frames, and possibly also in the

residual block. To apply the DA-PBT to the inter residual, instead of considering all the modes, the simplifying approach described in Section IV-C is again adopted. However, instead of measuring the directional gradients in the residual block that is to be transformed, the measurement is performed on the prediction block based on the assumption that the two blocks possess similar directionality if directional features still reside in the residual block. Only if the measured gradients satisfy the conditions in Section IV-C, suggesting high directionality in the prediction block and therefore possibly in the residual, one directional mode of the DA-PBT is considered in the direction selection process at the encoder, in addition to the nondirectional mode. At the decoder, these conditions can be examined by performing the same measurement on the prediction block generated during the decoding process. Therefore, only if the conditions are satisfied the decoder needs to decide whether the nondirectional mode or the directional mode has been selected, which can be signaled with a binary symbol from the encoder. More importantly, without further overhead, the decoder can recover this directional mode as it is the one with the smallest measured gradient. When applying the DA-PBT to inter residual blocks that already require less rate to encode, this approach greatly reduces the rate overhead that can easily nullify the potential gain of the DA-PBT.

VI. Experimental Results

The performance of the DA-PBT for both still image coding and video coding is reported in this section.

A. Still Image Coding

In the experimental results, the entropy coder described in Section IV-B, the adaptive deblocking filter in Section IV-A and the simplified search in Section IV-C are applied whereas the quantization matrices in Section III-C are disabled unless specifically mentioned.

We first present results using the 512×512 grayscale test images shown in Fig. 7. The rate-distortion performance of using various block-transform schemes for image coding with only 8×8 blocks is plotted in Fig. 8, including (1) **JPEG**: baseline JPEG with entropy coding defined in [1], without quantization matrices and with the deblocking filter in Section IV-A, (2) **DCT8**: the 2-D DCT, (3) **DDCT8**: the DDCT in [11], (4) **DDCT8+**: the DDCT with modified encoding orders of the transform coefficients, further explained in the next paragraph, (5) **DA-BT8** : the direction-adaptive transform described in Section III-A without partitioning, (6) **DA-PBT8** , and (7) **DA-PBT8-full**: the DA-PBT not using the simplified search in Section IV-C, i.e., searching through all nine modes. All schemes except **JPEG** are based on a common implementation, sharing the same entropy coder and deblocking filter. The only differences are in the transform itself and the encoding orders. In Fig. 8, the rates on the curves include the overheads signaling the selected blocksizes and modes, which are also indicated on the left of each plot for different schemes and reconstruction qualities.

To describe the encoding orders in **DDCT8+**, we use the three modes presented in Fig. 1 for 4×4 blocks as examples. For the vertical mode and the diagonal-down-right mode, instead of using the original encoding orders in Fig. 1(1a) and (1b), the orders in the corresponding modes in the DA-BT are adopted [Fig. 1(2a) and (2b)], making the two schemes equivalent in these modes. For the vertical-right mode, the encoding order starts with $\{r_1^{1\sim5}\}$ in Fig. 1(1c), followed by $\{r_2^{1\sim5}\}$, $\{r_3^{1\sim3}\}$, and finally $\{r_4^{1\sim3}\}$, based on the same arguments for the DA-BT in Section III-A.

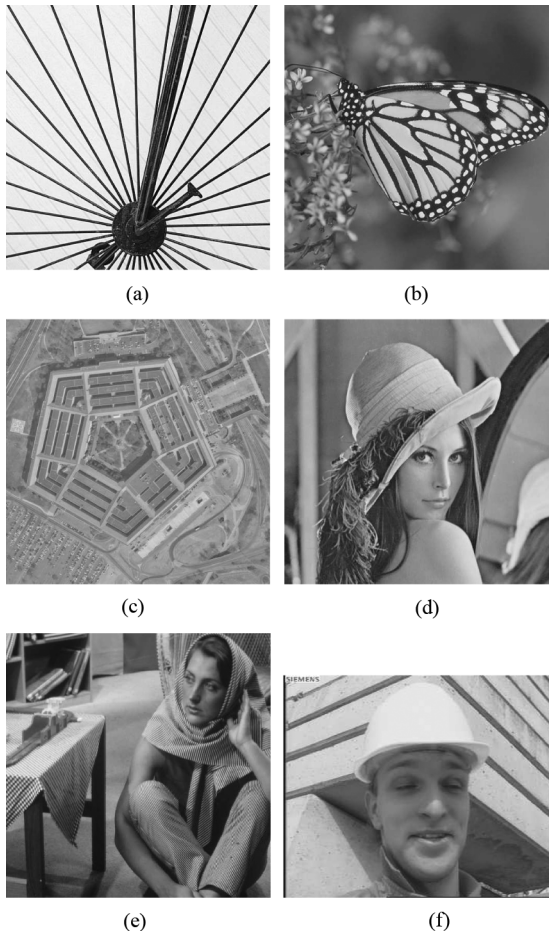

Fig. 7. Test images: (a) *Spoke*, (b) *Monarch*, (c) *Pentagon*, (d) *Lena*, (e) *Barbara*, and (f) *Foreman*.

From Fig. 8, it is clear that **DCT8** outperforms **JPEG** because of the more efficient entropy coder. Compared to **DCT8**, **DDCT8** only brings limited gain, which can generally be increased by the modified encoding orders in **DDCT8+**. Note that a part of this additional gain in **DDCT8+** comes from the vertical mode and the horizontal mode, which are, unlike in **DDCT8**, different from the nondirectional mode. **DA-BT8** further improves the performance upon **DDCT8+**, due to the modification of the transform directions in the vertical-right mode, the vertical-left mode, the horizontal-up mode, and the horizontal-down mode. Compared to **DCT8**, **DA-BT8** can improve the quality by more than 2 dB. However, the improvement in PSNR is less eminent for images with fewer directional features such as *Lena*. **DA-PBT8** keeps improving the performance in general by spatially confining energy within a partition while inducing less complexity (Section III-B). One exception is observed in *Barbara* where the stripes widespread in the image typically appear in all partitions in an 8×8 block. Additionally,

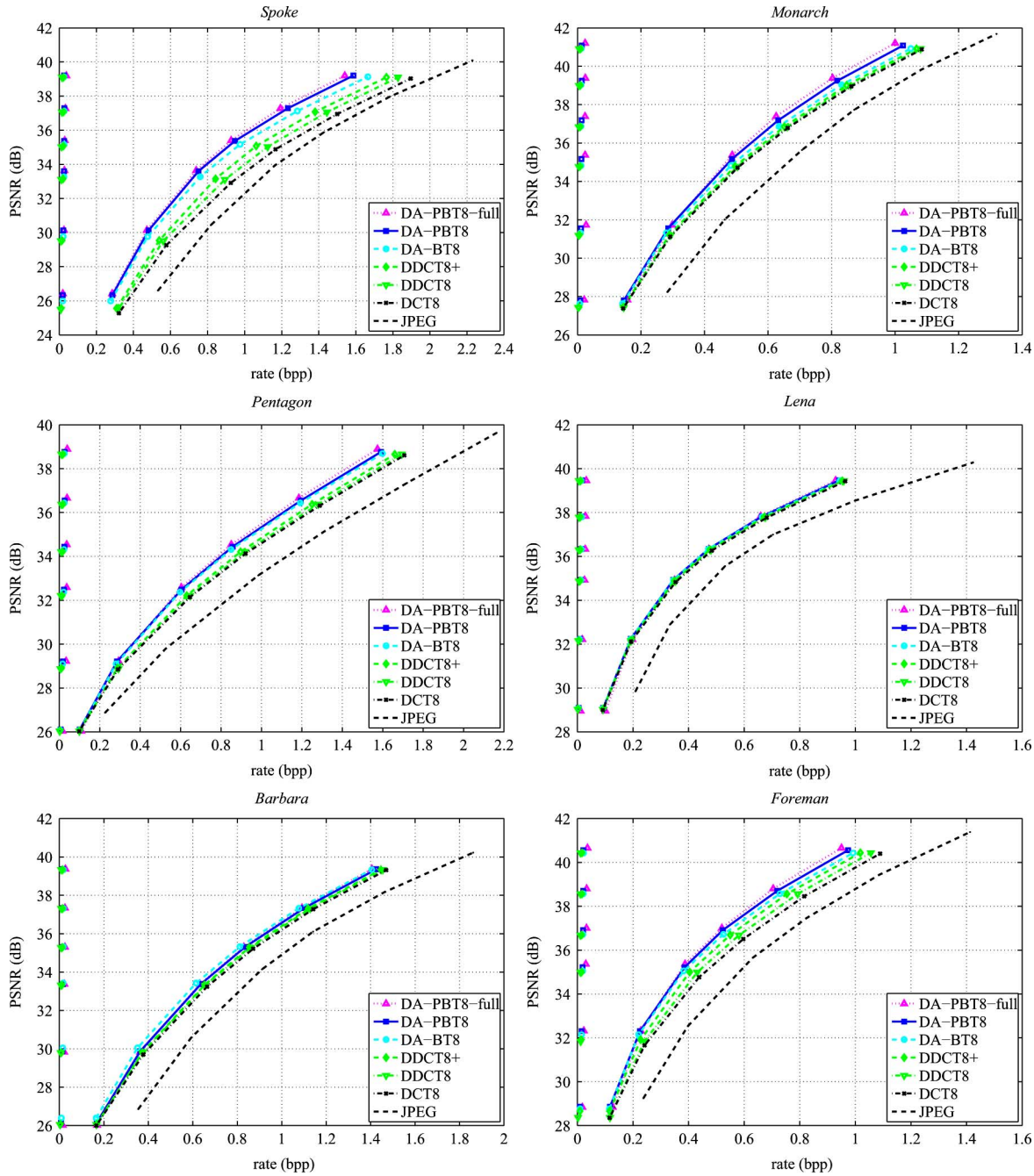

Fig. 8. Rate-distortion performance of 8×8 transforms for image coding. The rates on the curves include the signaling overheads, which are also indicated on the left of each plot for different schemes and reconstruction qualities.

the energy in these periodic patterns is very localized in frequency domain and can therefore be concentrated in fewer coefficients with a longer 1-D transform. Hence, in this special case the un-partitioned DA-BT outperforms the DA-PBT. Finally, the gap between **DA-PBT8-full** and **DA-PBT8** indicates the performance loss due to the simplified search, which is usually negligible considering the significant reduction in complexity.

Fig. 9 further includes the rate-distortion performance of variable blocksize transforms and directional intra prediction. **DCT8** again denotes the 8×8 2-D DCT, **DCT** and **DA-PBT** denote the 2-D DCT and the DA-PBT with variable blocksizes (4×4, 8×8, and 16×16), respectively, and **IAP(DCT)** and **IAP(DA-PBT)** denote directional intra prediction, again with variable blocksizes, together with the 2-D DCT and the DA-PBT applied to the prediction residual, respectively (Section V-A). From Fig. 9, **DCT** typically provides limited gain over **DCT8** except for *Monarch* where the energy in the low-frequency content in the blurred background can be better concentrated with larger 16×16 blocks. In addition, **DA-PBT** and **IAP(DA-PBT)** outperform **DCT** and **IAP(DCT)**, respectively, and the gain generally increases with the number of sharp directional features in the image. The gap between **IAP(DA-PBT)** and **IAP(DCT)** is usually smaller than that between **DA-PBT** and **DCT**. This is because in many cases the

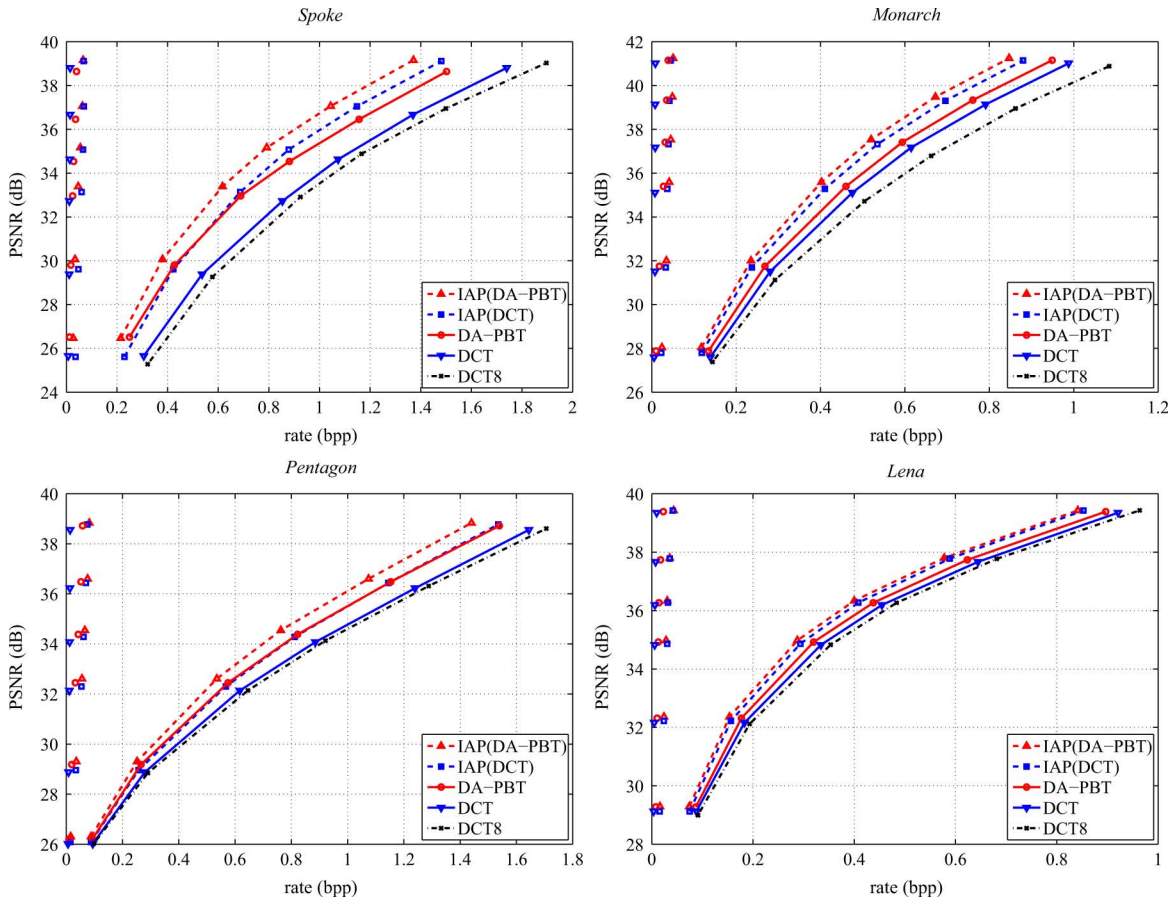

Fig. 9. Rate-distortion performance of variable blocksize transforms for image and residual image coding. The rates on the curves include the signaling overheads, which are also indicated on the left of each plot for different schemes and reconstruction qualities.

residual energy is small so that the transform coefficients are all quantized to zero regardless of which transform is applied, leading to the same rate and distortion. As an extreme example, with perfect intra prediction the residual is zero and the transform does not at all affect the coding performance. Comparing **IAP(DCT)** and **DA-PBT** , directional intra prediction is usually more efficient than the directional transforms for being able to utilize the correlation across block boundaries. However, compared to the prediction-based approach, the transform-based approach possesses two main advantages. First, with a proper design of entropy coding and bitstream organization, each macroblock (or 4×4 macroblocks when an additional 4×4 transform is applied as in the proposed approach described in Section IV-A) can be decoded independently from others, providing better support for random access and error resiliency. Second, both block-wise and image-wise, the transform is close to orthonormal, allowing embedded (quality-progressive) coding of images [25], [26].

To demonstrate the improvement in visual quality using the DA-PBT, a 256×256 region of *Pentagon* is shown in Fig. 10(a), and the corresponding reconstructions from **DCT**, **DA-PBT**, **IAP(DCT)**, and **IAP(DA-PBT)** using the same quantization stepsize are shown in Fig. 10(b)–(e). The DA-PBT , applied both to the image and to the intra residual, greatly reduces the ringing and checkerboard artifacts around edges observed in the DCT-based schemes while demanding less rates. **DCT** with a rate comparable to that in **IAP(DA-PBT)** [Fig. 10(e)] is shown in Fig. 10(f), demonstrating the advantages of exploiting directionality in the images for image compression, both in the prediction and in the transform. Similar observations can be made for *Monarch* as shown in Fig. 11.

To show the effect of the quantization matrices and the adaptive deblocking filter, the reconstructions of the *Sailboat* color image using the DA-PBT are included in Fig. 12. Comparing Fig. 12(b) and (c), both without the deblocking filter, using the quantization matrices avoids spending bits on refining the complex high-frequency patterns around the waves at the bottom of the image where the refinement is generally not perceivable. Instead, more bits can be allocated to better represent the low-frequency content, especially the DC levels, and the more structural features. This is demonstrated in Fig. 12(c) by the reduced blocking artifacts and the better preserved fine lines on the sails. The blocking artifacts in both Fig. 12(b) and (c) can be largely removed by the adaptive deblocking filter as shown in Fig. 12(d) and (e). Due to the heavy blocking artifacts originally in Fig. 12(b), certain areas in Fig. 12(d) still appear blocky whereas Fig. 12(e) retains more details and overall a better quality.

B. Video Coding

For video coding, as a proof of concept, we implemented a video coder to evaluate the potential gain from the DA-PBT

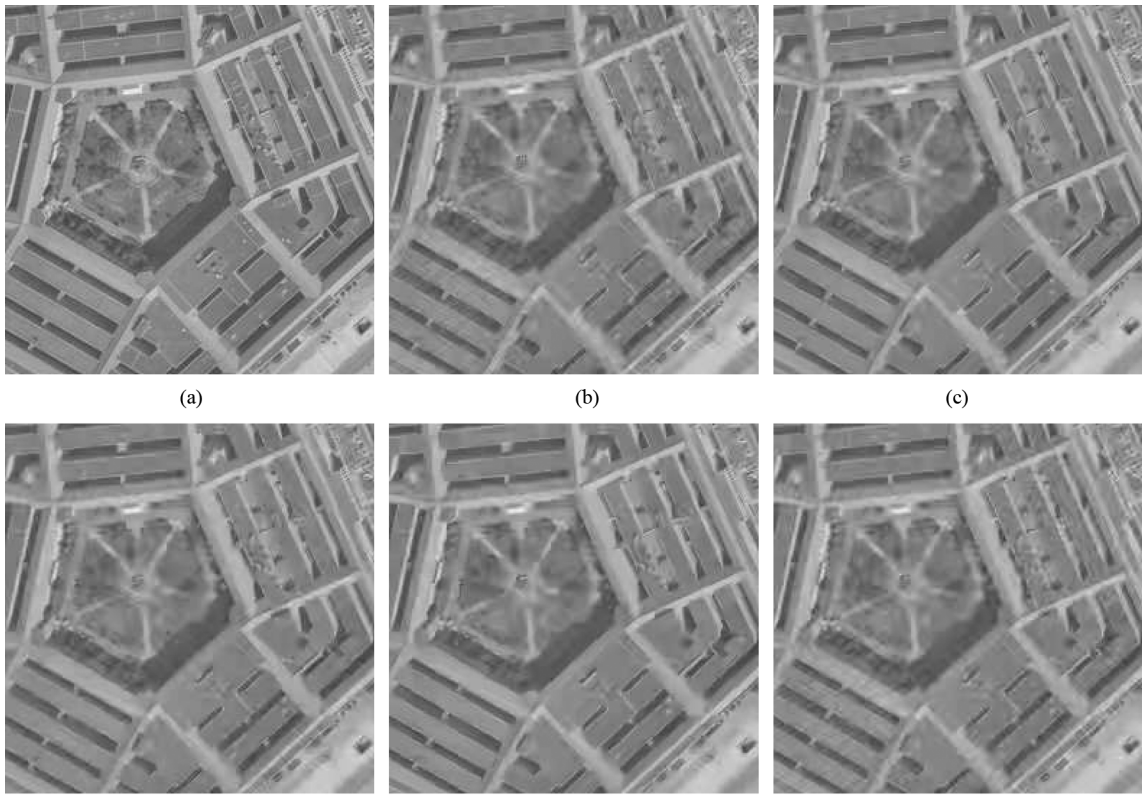

Fig. 10. Reconstruction of a 256 × 256 region in the *Pentagon* image. (a) Original, (b) DCT (0.28 bpp, 28.88 dB), (c) DA-PBT (0.27 bpp, 29.18 dB), (d) IAP(DCT) (0.26 bpp, 28.96 dB), (e) IAP(DA-PBT) (0.25 bpp, 29.31 dB), and (f) DCT (0.25 bpp, 28.52 dB).

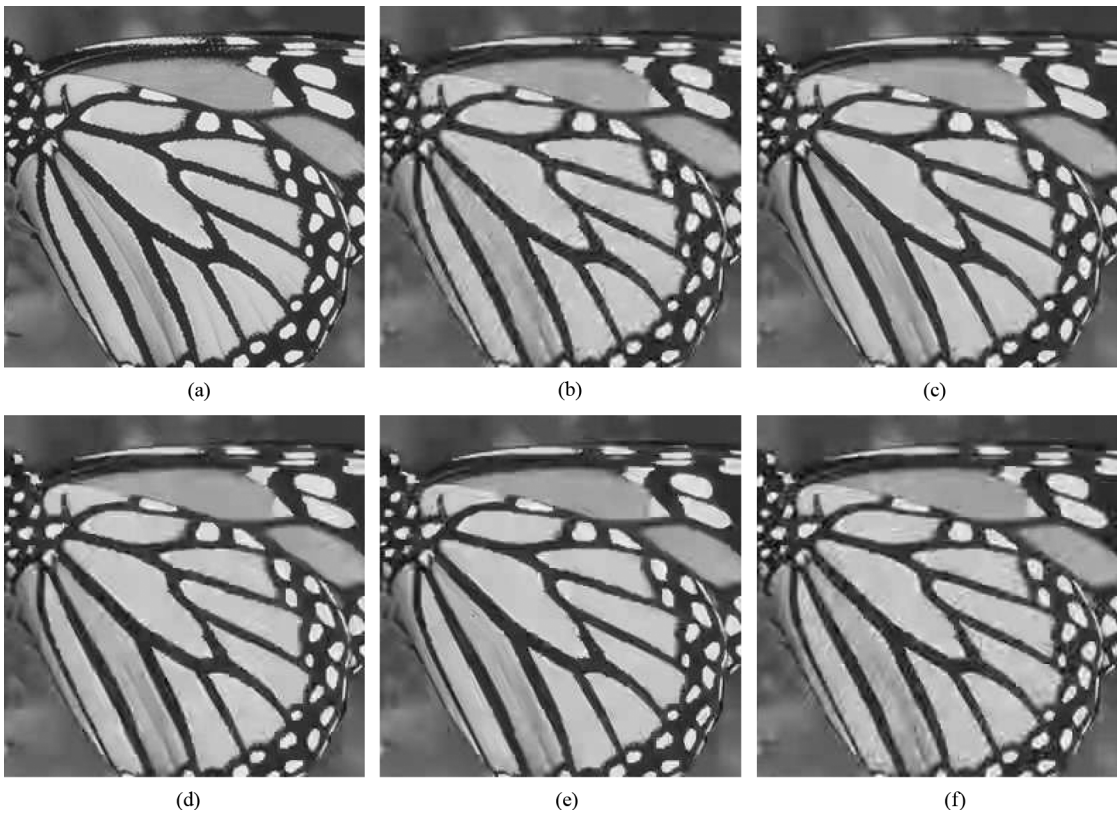

Fig. 11. Reconstruction of a 256 × 256 region in the *Monarch* image. (a) Original, (b) DCT (0.17 bpp, 28.44 dB), (c) DA-PBT (0.16 bpp, 28.72 dB), (d) IAP(DCT) (0.14 bpp, 28.64 dB), (e) IAP(DA-PBT) (0.14 bpp, 28.90 dB), and (f) DCT (0.14 bpp, 27.60 dB).

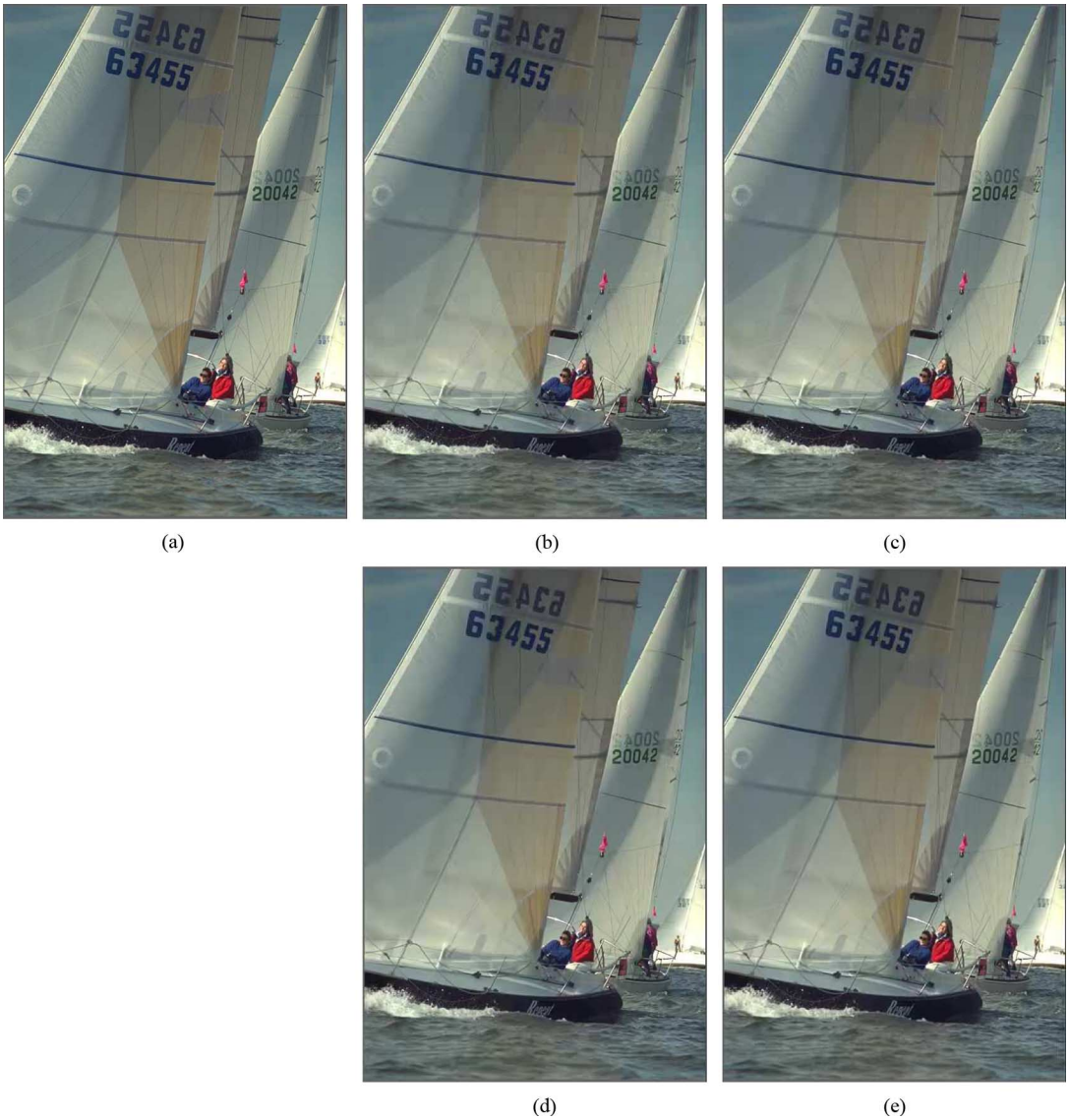

Fig. 12. Reconstruction of the 512×768 *Sailboat* image using the DA-PBT at around 0.36 bpp, with or without the quantization matrices (QM) and the deblocking filter (DB). (a) Original, (b) DA-PBT, w/o QM, w/o DB, (c) DA-PBT, with QM, w/o DB, (d) DA-PBT, w/o QM, with DB, and (e) DA-PBT, with QM, with DB.

when it is incorporated into state-of-the-art video algorithms. For simplicity, we consider only the luminance component. The video frames are divided into I-pictures and P-pictures, and the 16×16 macroblocks in these pictures are categorized into three modes, INTRA, INTER, and SKIP, similar to those defined in H.264 [4]. The macroblocks in the I-pictures are all in the INTRA mode and are encoded using either **IAP(DCT)** or **IAP(DA-PBT)** as described above for still images. For the P-pictures, each macroblock can be in one of the three modes. The INTRA macroblocks are again encoded by **IAP(DCT)** or **IAP(DA-PBT)**. The INTER macroblocks are first predicted with motion-compensated inter prediction using 4×4, 8×8, or 16×16 blocks, and the residual blocks are encoded using either the 2-D DCT or the DA-PBT, denoted by **IRP(DCT)** and **IRP(DA-PBT)**, respectively. A SKIP macroblock directly copies a macroblock from the reconstruction of the previous frame using motion compensation with a motion vector derived from the causal neighborhood [4]. Motion-compensated inter prediction is realized similarly to H.264 with quarter-pel accuracy and a search range of $\pm 32 \times \pm 32$ pixels. To quantize the transform coefficients, the quantization parameter $Q_{H.264}$ (Section IV-A) for the P-pictures is set to be one more than that for the I-pictures, and the parameter Δ (Section III-C) is set to $1/3$ for INTRA and $1/6$ for INTER macroblocks [18]. The quantization indices and the overhead signaling all the mode and blocksize selections are encoded using CABAC-like entropy coding as discussed in Section IV-B, whereas the motion vectors are first predicted from causal neighborhoods and then encoded using a fixed variable-length-coding table.

We consider two coding arrangements: all I-pictures and one I-picture in every 15 frames. The performance of the corresponding baseline method using the 2-D DCT is denoted by **IAP(DCT)** in Fig. 13 and **IAP(DCT)+IRP(DCT)** in Fig. 14, respectively, for the CIF sequences *Foreman* and *Carphone*. To benchmark the implemented baseline methods, the rate-distortion performance of the VCEG KTA reference software [27], an extension of the JM reference software of H.264 [28], is included, denoted by **KTA I15** in Fig. 13 and **KTA IP14** in

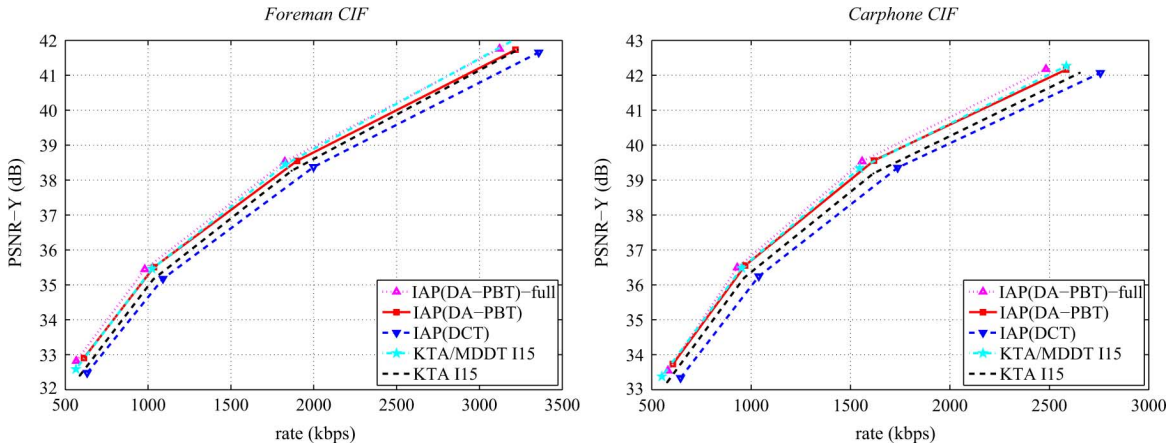

Fig. 13. Rate-distortion performance of using the 2-D DCT and the DA-PBT for video coding with all I-pictures.

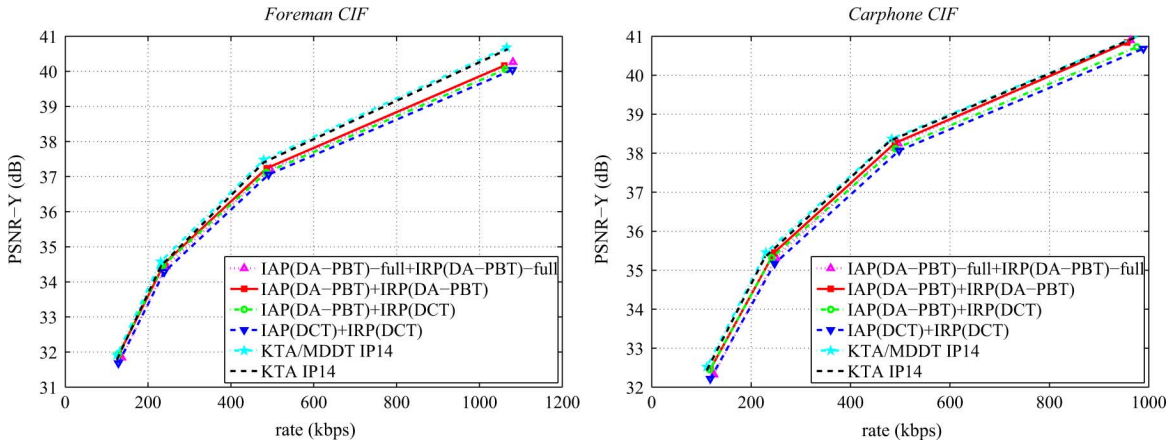

Fig. 14. Rate-distortion performance of using the 2-D DCT and the DA-PBT for video coding with one I-picture followed by 14 P-pictures.

Fig. 14 for the two arrangements, respectively, using the High Profile (FREext) that additionally enables 8×8 intra prediction and transforms [29]. To be more consistent with the configuration implemented in our coder, in the KTA reference software, only one reference frame is used, B-frames, motion compensation with rectangular blocks, adaptive rounding and the IPCM mode are all disabled. The other encoding parameters remain the default values [27]. Note that unlike the integer transforms adopted in H.264 [4], the transforms in our coder use floating-point operations. An integer DA-PBT may be constructed by techniques for designing integer DCTs [30], but requires further investigation. Additionally, H.264 defines only four intra prediction modes for 16×16 blocks and the associated transform is accomplished by sixteen 4×4 transforms, whereas there are nine modes in our implementation and the transform is a 16×16 DCT .

From Figs. 13 and 14, our baseline coder is less efficient than the KTA reference software due to several simplifications in the implementation. For instance, the High Profile in H.264 allows the 4×4 or the 8×8 transform to be selected adaptively in INTER macroblocks [29], whereas in our implementation the size of the transform is coupled to the blocksize used in motion-compensated prediction. Nonetheless, the performance of our baseline coder is still close to that of the KTA reference software, suggesting that it is an adequate implementation of state-of-the-art video coding algorithms, sufficient for further evaluation of the effectiveness of the DA-PBT. The KTA reference software also includes an option that uses the mode-dependent directional transform (MDDT) for INTRA macroblocks [31] based on the work later presented in [13]. The performance of KTA using the MDDT is included for comparison, denoted by **KTA/MDDT I15** in Fig. 13 and **KTA/MDDT IP14** in Fig. 14. Note that for 16×16 blocks, different from using sixteen 4×4 transforms in H.264, the MDDT and the DA-PBT both use actual 16×16 transforms.

For using all I-pictures, compared to **IAP(DCT)**, **IAP(DA-PBT)** on average improves the reconstruction quality by 0.51 dB for *Foreman* and 0.66 dB for *Carphone*, and, equivalently, reduces the rate by 9.0% and 10.5%, respectively, as shown in Fig. 13. Furthermore, instead of using the same mode for the DA-PBT and directional intra prediction, the performance of considering all of the nine DA-PBT modes regardless the intra prediction mode selected is also included in the figure, denoted by **IAP(DA-PBT)-full**. The performance loss from using the same mode is generally acceptable considering the additional computation required to evaluate the rate and distortion associated with all the other DA-PBT modes. Comparing the gap between **IAP(DA-PBT)** and **IAP(DCT)** and that between **KTA/MDDT I15** and **KTA I15**, the DA-PBT in general brings more gain than the MDDT upon their nondirectional counterparts,

suggesting potential improvement of the KTA reference software using the DA-PBT.

For the arrangement with inter prediction, the DA-PBT can be applied to the INTRA macroblocks, denoted by **IAP(DA-PBT)+IRP(DCT)**, or additionally to the INTER macroblocks, denoted by **IAP(DA-PBT)+IRP(DA-PBT)** in Fig. 14. Note that the KTA reference software using the MDDT, shown as **KTA/MDDT IP14** in Fig. 14, only applies directional transforms to the INTRA macroblocks. As discussed in Section V-B, **IRP(DA-PBT)** considers at most one directional mode of the DA-PBT . The scheme that enables all nine modes of the DA-PBT to be selected for both the intra and the inter residual blocks is denoted by **IAP(DA-PBT)-full+IRP(DA-PBT)-full** and is also included in the figure. Although **IAP(DA-PBT)-full** outperforms **IAP(DA-PBT)** in Fig. 13, considering all DA-PBT modes for the inter residual as in **IAP(DA-PBT)-full+IRP(DA-PBT)-full** surprisingly deteriorates the performance. Despite the additional flexibility, **IAP(DA-PBT)-full+IRP(DA-PBT)-full** no longer achieves better rate-distortion performance than **IAP(DA-PBT)+IRP(DA-PBT)** because of the increased overhead for the INTER macroblocks that offsets the potential gain (Section V-B). On average, compared to **IAP(DCT)+IRP(DCT)**, **IAP(DA-PBT)+IRP(DA-PBT)** improves the quality by 0.22 dB for *Foreman* and 0.27 dB for *Carphone*, and equivalently reduces the rate by 5.5% and 6.6%, respectively. Similar to comparing the effectiveness of the DA-PBT on image blocks and intra residual blocks, the DA-PBT is less effective on inter than intra residual since inter prediction is typically more accurate and leads to smaller residual energy. This contradicts the observation in [11] where the DDCT is more effective on inter residual blocks rather than on image blocks. We conjecture that this is due to the less accurate motion compensation adopted in [11] that always uses 16×16 blocks and a search range of mere $\pm 7 \times \pm 7$ pixels.

VII. Conclusion

We have proposed a new direction-adaptive partitioned block transform (DA-PBT) and explored its effectiveness for coding of color images and video sequences. The DA-PBT outperforms the conventional 2-D DCT by more than 2 dB for images with pronounced directional features. Since it avoids the typical ringing and checkerboard artifacts of the conventional 2-D DCT, subjective gains are even larger than indicated by rate-PSNR performance. The DA-PBT also outperforms a previously proposed directional DCT while demanding less computation, owing to an improved directional selectivity, direction-adaptive partitioning and improved coefficient ordering for entropy coding. As for conventional block transforms, the DA-PBT achieves its best performance when combined with adaptive blocksizes. To exploit the frequency response of the human visual system, quantization stepsize matrices should be used with a directional block transform. We show a straightforward way to transform JPEG quantization matrices into the directional transform space, thus, avoiding cumbersome subjective tests.

We have also explored the combination of the DA-PBT with predictive coding, either directional prediction within the image or interframe prediction for video. For intraframe directional prediction, the direction selection for the prediction and the transform can be elegantly combined. It is not surprising that the gains of both techniques are not additive, as similar signal properties are exploited by the prediction and the transform. Since the DA-PBT operates in a block-wise manner, the incorporation into block-based motion-compensated video coding is straightforward. Alas, we have not been able to demonstrate significant gains by compressing the motion-compensated prediction residual with the DA-PBT when advanced motion compensation is used for our relatively simple test sequences. The DA-PBT might still have a role to play for video coding, at least for intracoded blocks, or where motion-compensated prediction is not fully effective.

References

[1] *Recommendation T.81, Information Technology: Digital Compression and Coding of Continuous-Tone Still Images*, ISO/IEC 10918–1 ITU-T, 1992.

[2] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2002.

[3] *Information Technology: JPEG 2000 Image Coding System*, ISO/IEC 15444–1:2000, 2002.

[4] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[5] P. Pirsch, "Stability conditions for DPCM coders," *IEEE Trans. Commun.*, vol. COM–30, pp. 1174–1184, May 1982.

[6] D. Wang, L. Zhang, A. Vincent, and F. Speranza, "Curved wavelet transform for image coding," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2413–2421, Aug. 2006.

[7] 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.

[8] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform for image compression," *IEEE Trans. Image Process.*, vol. 16, pp. 1289–1302, May 2007.

[9] V. Velisavljevic, B. Beferull-Lozano, and M. Vetterli, "Space-frequency quantization for image compression with directionlets," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1761–1773, Jul. 2007.

[10] B. Zeng and J. Fu, "Directional discrete cosine transforms for image coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, Toronto, Ontario, Canada, Jul. 2006, pp. 721–724.

[11] 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.

[12] 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.

[13] 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 Int. Conf. Image Processing*, San Diego, CA, Oct. 2008, pp. 2116–2119.

[14] C.-L. Chang and B. Girod, "Direction-adaptive partitioned block transform for image coding," in *Proc. IEEE Int. Conf. Image Processing*, San Diego, CA, Oct. 2008, pp. 145–148.

[15] C.-L. Chang, "Direction-Adaptive Transforms for Image Communication," Ph.D, Stanford University, Stanford, CA, 2009.

[16] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[17] P. Kauff and K. Schuur, "A shape-adaptive DCT with block-based DC separation and $\triangle$DC correction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 237–242, Jun. 1998.

[18] T. Wedi and S. Wittmann, "Quantization offsets for video coding," in *IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 2005, pp. 324–327.

[19] H. Lohscheller, "A subjectively adapted image communication system," *IEEE Trans. Commun.*, vol. COM–32, no. 12, pp. 1316–1322, Dec. 1984.

[20] A. J. Ahumada, Jr and H. A. Peterson, "A visual detection model for DCT coefficient quantization," in *Proc. 9th AIAA Comput. Aerospace Conf.*, San Diego, CA, Oct. 1993, pp. 314–318.
[21] T. Wiegand, H. Schwarz, A. Joch, and F. Kossentini, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.
[22] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. IEEE Int. Conf. Image Process.*, Thessaloniki, Greece, Oct. 2001, vol. 3, pp. 542–545.
[23] M. Wien, "Variable block-size transforms for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 604–613, Jul. 2003.
[24] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.
[25] Z. Xiong, O. G. Guleryuz, and M. T. Orchard, "A DCT-based embedded image coder," *IEEE Signal Process. Lett.*, vol. 3, no. 11, pp. 289–290, Nov. 1996.
[26] V. P. Shah, J. E. Fowler, and N. H. Younan, "Tarp filtering of block-transform coefficients for embedded image coding," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, Toulouse, France, May 2006, p. II.
[27] VCEG KTA Reference Software Sep. 2008 [Online]. Available: http://iphome.hhi.de/suehring/tml/download/KTA/jm11.0kta2.1.zip
[28] H.264/MPEG-4 AVC Reference Software Manual (JVT-X072), Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG 2007.
[29] D. Marpe, T. Wiegand, and S. Gordon, "H.264/MPEG4-AVC fidelity range extensions: Tools, profiles, performance, and application areas," in *Proc. IEEE Int. Conf. Image Processing*, Genova, Italy, 2005, pp. 593–596.
[30] 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. 2001.
[31] *Improved intra coding*, ITU-T Q.6/SG16 VCEG-AF15, 2007.

Chuo-Ling Chang (S'03–M'09) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1998 and the M.S. and Ph.D. degrees in electrical engineering from the Information Systems Laboratory of Stanford University, Stanford, CA, in 2002 and 2009, respectively.

Since 2008, he has been working with several startup companies, developing novel communication systems for live video sharing and video conferencing. He is currently a Senior Audio/Video Scientist at TokBox Inc., San Francisco, CA. His research interests mainly include scalable coding and streaming of multimedia data.

Mina Makar (S'09) received the B.S. and M.S. degrees in electrical engineering from Alexandria University, Egypt, in 2004 and 2006, respectively, and is currently pursuing the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

In 2007, he joined the Electrical Engineering Department at Stanford University. His research interests include video coding and streaming for interactive video delivery.

Sam S. Tsai (S'09) received the B.S. and M.S. degrees from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 2001 and 2003, respectively, and is currently pursuing the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

From 2003 to 2007, he worked at Realtek Semiconductor Corp. developing digital multimedia systems as an engineer. His research interest includes image and video compression, mobile multimedia systems, and mobile image retrieval.

Bernd Girod (M'80–SM'97–F'98) received the M.S. degree from the Georgia Institute of Technology, Atlanta, and the Engineering Doctorate from the University of Hannover, Hannover, Germany.

He has been a Professor of electrical engineering and (by courtesy) computer science in the Information Systems Laboratory of Stanford University, Stanford, CA, since 1999. Previously, he was a Professor of telecommunications in the Electrical Engineering Department of the University of Erlangen-Nuremberg. His current research interests are in the areas of video compression and networked media systems. He has published over 400 conference and journal papers, as well as five books. As an entrepreneur, he has been involved with several startup ventures as founder, director, investor, or advisor, among them Polycom (Nasdaq: PLCM), Vivo Software, 8x8 (Nasdaq: EGHT), and RealNetworks (Nasdaq: RNWK).

Prof. Girod is a EURASIP Fellow and a member of the German National Academy of Sciences (Leopoldina). He received the EURASIP Signal Processing Best Paper Award in 2002, the IEEE Multimedia Communication Best Paper Award in 2007, the EURASIP Image Communication Best Paper Award in 2008, as well as the EURASIP Technical Achievement Award in 2004