Video Coding with Rate-Distortion Optimized Transform

Xin Zhao, Li Zhang, Siwei Ma, *Student Member, IEEE,* and Wen Gao, *Fellow, IEEE*

***Abstract*—Block-based discrete cosine transform (DCT) has been successfully adopted into several international image/video coding standards, e.g., MPEG-2, H.264/AVC, as it can achieve a good tradeoff between performance and complexity. Although DCT theoretically approximates the optimum Karhunen–Loève transform under first-order Markov conditions, one fixed set of transform basis functions (TBF) cannot handle all the cases efficiently due to the non-stationary nature of video contents. To further improve the performance of block-based transform coding, in this paper, we present the design of rate-distortion optimized transform (RDOT) which contributes to both intraframe and interframe coding. The most important property which makes a difference between RDOT and the conventional DCT is that, in the proposed method, transform is implemented with multiple TBF candidates which are obtained from off-line training. With this feature, for coding each residual block, the encoder is capable to select the optimal set of TBF in terms of rate-distortion performance, and better energy compaction is achieved in the transform domain. To obtain an optimum group of candidate TBF, we have developed a two-step iterative optimization technique for the off-line training, with which the TBF candidates are refined at each iteration until the training process becomes converged. Moreover, analysis on the optimal group of candidate TBF is also presented in this paper, with a detailed description of a practical implementation for the proposed algorithm on the latest VCEG key technical area software platform. Extensive experimental results show that, compared with the conventional DCT-based transform scheme adopted into the state-of-the-art H.264/AVC video coding standard, significant improvement of coding performance has been achieved for both intraframe and interframe coding with our proposed method.**

***Index Terms*—Directional transform, H.264/AVC, Karhunen–Loève transform (KLT), mode-dependent directional transform (MDDT), video coding.**

I. Introduction

BECAUSE OF ITS outstanding energy compaction ability [1], transform coding has become a successful method in the international video coding standards. With transform coding, correlations of signal existing in the spatial domain are efficiently removed in the transform domain, which is of great importance for the subsequent processing including coefficient scanning and entropy coding. Among all the well-known transforms developed in the past several decades, DCT has become a particular favor for natural imagery sources because it was both theoretically proved and experimentally confirmed that DCT approximates the optimum transform [Karhunen–Loève transform (KLT)] [4], [5] in the sense of energy compaction under the first-order Markov conditions [2], [3]. In the scenario of image and video coding, a 2-D variant of block-based DCT with a separable formulation has been widely employed in almost all transform-based international image and video coding standards, including JPEG [6], MPEG 1/2/4 [7]–[9], H.261 [10], H.263 [11], H.264/AVC [12], and AVS [13].

Advances of block-based transform coding in the literature mainly concentrate on two motivations: lower complexity and higher performance. For the first motivation, a fast DCT algorithm was proposed in [14] by factorizing the coefficient matrix into products of simpler matrices. Furthermore, to implement DCT with fixed-point arithmetic, integer cosine transform (ICT) was introduced in [15] with negligible degradation of the transform efficiency. With the development of ICT, difficulties regarding the storage of irrational basis functions and expensive calculation of floating-point arithmetic are solved, and the inverse transform mismatch problems are also perfectly handled. Moreover, a low-complexity 4×4 transform, with only adds and shifts in 16-bit arithmetic, was proposed in [16] as an important contribution to the H.264/AVC standard.

Efforts devoted to the second motivation are focused on a further promotion of transform efficiency or a better reduction of visual artifacts, e.g., blocking artifacts. In general, 2-D transform is usually implemented by two separable 1-D transforms, which extract the signal correlation along the horizontal and vertical directions, respectively. With this separable form, 2-D transform becomes a simple task by doing 1-D transform twice, each along horizontal/vertical direction. However, for natural imagery sources where edges can be arbitrarily directed, the conventional 2-D DCT becomes inefficient in the sense of energy compaction. This inefficient representation of edges is not desirable for compression because it may result in lots of bits for the expensive coding of high-frequency coefficients. Moreover, annoying blocking and ringing artifacts can be observed when these high-frequency coefficients are strongly quantized at low bit rates [22].

Manuscript received September 10, 2010; revised December 29, 2010 and March 8, 2011; accepted May 1, 2011. Date of publication May 31, 2011; date of current version January 6, 2012. This work was supported in part by the National Science Foundation of China, under Grants 60833013 and 60803068, and in part by the National Basic Research Program of China (973 Program), under Grants 2009CB320903 and 2009CB320904. This paper was recommended by Associate Editor R. Rinaldo.

X. Zhao is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China (e-mail: xzhao@jdl.ac.cn).

L. Zhang, S. Ma, and W. Gao are with the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: li.zhang@pku.edu.cn; swma@pku.edu.cn; wgao@pku.edu.cn).

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.2011.2158363

For the second motivation, new transform schemes which present better performance than the conventional DCT have been developed in the literature. A noteworthy advance in this research field is the exploration of new transforms with directional wavelet bases. The framework of directional filter-bank decomposition was introduced in [17], and developments of new directional wavelet bases include the invention of the so-called ridgelet transform [18], curvelet transform [19], and contourlet transform [20]. To incorporate the idea of using directional bases into the classical block-based hybrid video coding scheme, a directional extension of conventional DCT was proposed in [21]. Rather than the horizontal and vertical directions in conventional 2-D DCT, the proposed directional DCT employs two 1-D DCTs along other directional orientations. With the proposed directional DCT, arbitrarily directed edges are better accommodated than the conventional DCT, and superior coding performance has been validated for H.264/AVC intracoding. Furthermore, in intracoding where the residual samples within a single block present different correlation along different directions in directional prediction, KLT achieves noticeable improvement over conventional DCT with the data-dependent basis functions. In recent years, KLT-based transform has been employed to further improve the H.264/AVC intracoding. Characterized by the fact that the residual samples present different statistics for different intraprediction (IP) modes, mode-dependent directional transform (MDDT) is proposed in [23], i.e., different transform functions are employed for different IP modes. A new transform scheme which adaptively employs integer version of DCT or discrete sine transform for prediction residue is also proposed in [24], and the various statistics of residual samples are better accommodated. Moreover, a novel rate-distortion optimized transform (RDOT) scheme based on multiple transform function candidates is recently proposed in [25] for intracoding. The transform function candidates are obtained offline using a two-step iterative training method, and superior coding performance has been achieved compared to MDDT.

The remainder of this paper is organized as follows. In Section II, a brief review of the directional DCT and MDDT for intracoding is provided. The technical details of the proposed RDOT and its fast implementation are introduced in Section III, and a brief theoretical analysis on the training of an optimal group of transform function candidates is presented in Section IV. Section V shows the extensive experimental results and analysis to validate the performance of the proposed method. Finally, this paper is concluded in Section VI with a brief description of some expected future work.

II. Brief Review of Directional DCT and MDDT

In this section, a brief review of two recent directional transforms including directional DCT and MDDT is presented. Both two transforms are proposed within the framework of block-based image and video coding and target at a further promotion of coding performance over the conventional DCT.

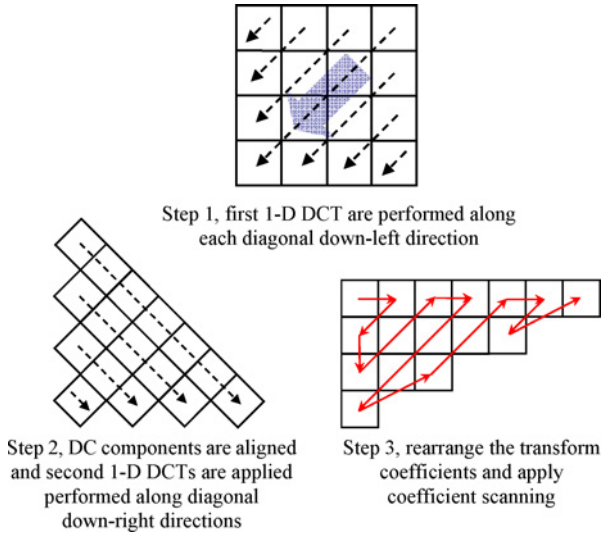

Fig. 1. Transform a residual block in diagonal down-left intramode with directional DCT [21].

A. Directional Discrete Cosine Transform

As depicted earlier, conventional 2-D DCT is implemented separately by performing 1-D DCT twice, horizontally and vertically. Naturally, this separable 2-D transform is capable to capture the signal correlation along either the horizontal or vertical direction. However, for natural images where edges can be flexibly directed, e.g., diagonal edges, the decorrelation performance of conventional scheme becomes sub-optimal. To resolve this problem, directional DCT is proposed in [21] for image coding.

To illustrate the basic idea of directional DCT, let us consider the diagonal down-left mode in H.264/AVC intracoding. As shown in Fig. 1 [21], the major modifications of conventional intracoding scheme lie in three conventional stages, including the horizontal transform, vertical transform, and coefficient scanning. At the first stage, 1-D DCT is performed along each of the diagonal down-left directed lines of residual samples, and the transform coefficients are placed in a non-rectangular shape due to the different transform sizes applied to each line of samples. Furthermore, for directional prediction (IP Mode "3"–"8" in H.264/AVC), a ΔDC correction process is implemented after the first 1-D DCT to handle the so-called mean weighting defect [26], which is caused by different transform sizes applied for different directional lines. After that, the DC components are aligned and a second 1-D DCT is applied along each of the diagonal down-right directed lines of coefficients. Finally, the transform coefficients are quantized, scanned, and entropy coded after the DC components are aligned, where the scanning orders are modified from the conventional zig-zag order as illustrated in Fig. 1.

B. Mode-Dependent Directional Transform

Two important observations on intraprediction residue motivate the basic idea of MDDT. First, for different intramodes, the energy of residual samples is distributed differently within the region of a single residual block. To verify this, a large number of the residual samples are collected from an actual

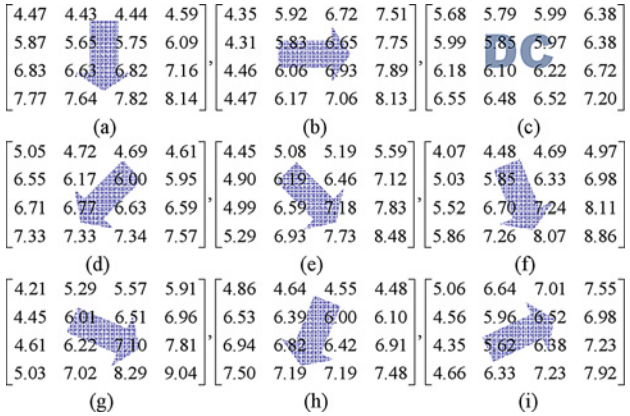

Fig. 2. Normalized distributions of absolute residue magnitudes in 4×4 intraprediction modes. (a) Mode 0. (b) Mode 1. (c) Mode 2. (d) Mode 3. (e) Mode 4. (f) Mode 5. (g) Mode 6. (h) Mode 7. (i) Mode 8.

H.264/AVC intracoding process, and the normalized distributions of absolute residue magnitudes in 4×4 IP modes are demonstrated in Fig. 2, where the broad arrows beneath the digits indicate the prediction directions. It is clearly observed that the absolute magnitudes of residual samples are position-dependent within a residual block. Second, it is noted that the residual samples present distinguishable distribution characteristics for different IP modes. For example, for IP Mode 0, the absolute magnitudes present similar values at horizontal direction, but increase along vertical direction. This is because for Mode 0 where vertical prediction is applied, the first few rows of residual samples are closer to the prediction pixels, therefore lower energy than the bottom few rows is present owing to the stronger correlation with the prediction pixels. Similarly, for the case of Mode 1, the absolute magnitudes present similar values at vertical direction, but increase along horizontal direction.

Based on the first observation, it is interesting to see that DCT no longer approximates the optimum transform for directional intraprediction residue, and KLT-based transform achieves noticeable improvements. This conclusion has also been analytically pointed out in [27] that for vertical prediction, the row-wise covariance matrix is not a Toeplitz matrix, and therefore DCT becomes sub-optimal. Motivated by the second observation, which reveals the distinguishable characteristics of residue distributions in different intramodes, mode-dependent transforms are proposed in MDDT to further refine the transform for intraprediction residues. Due to complexity related considerations, separable transforms are employed in MDDT with the following formulation:

$$F = C_i \cdot X \cdot R_i \tag{1}$$

where X indicates the residual block C_i and R_i are the vertical (column) and horizontal (row) transform functions for IP mode i, respectively, and F denotes the resulting transform coefficient matrix. Different from the conventional DCT, in MDDT, the column and row transform matrices are trained off-line with actual residual blocks generated by prediction mode i and are no longer the transposed version of each other. In MDDT, the components of both C_i and R_i are scaled by a factor of 2^7 and rounded as integers lying in the range of $(-128, 128)$. With the contribution of mode-dependent feature, MDDT efficiently improves the transform for intraprediction residue in H.264/AVC and was successfully adopted into the key technical area (KTA) software [32].

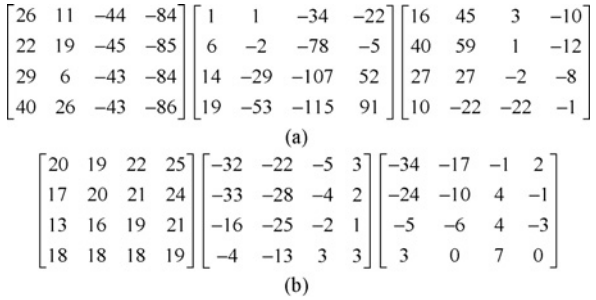

Fig. 3. Actual intrapredicted residual blocks with the same intraprediction mode 0, and interpredicted residual blocks. (a) Intrapredicted residual blocks. (b) Interpredicted residual blocks.

III. Proposed Rate-Distortion Optimized Transform

In this section, we start with several motivating observations which provide useful guidelines for introducing the proposed algorithm. Then the detailed implementation of the proposed rate-distortion optimized transform for both intra and intercoding is described, respectively. Furthermore, to reduce the high encoding complexity for practical applications, a fast RDOT scheme is introduced with significant reduction of overall encoding time, while the coding performance degradation is negligible.

A. Observations of Several Intra and Interpredicted Residual Blocks

Although MDDT efficiently improves the intratransform efficiency by employing different transform functions for different intraprediction modes, in our simulations, it is observed that even in the same mode, the residue always presents different statistical characteristics. To verify this case, several intra and interpredicted residual blocks are shown in Fig. 3. As it shows, all three 4×4 blocks in Fig. 3(a) are actual residual blocks obtained from the intra mode 0 (vertical prediction). However, the left block presents a vertical edge, but the other two blocks present irregular textures. Also for interpredicted blocks shown in Fig. 3(b), three blocks present distinctive characteristics.

The above observations imply the possibility to further improve the transform efficiency of both intra and interpredicted residual blocks, and it leads us to the idea of using multiple transform matrix candidates which is naturally capable to accommodate the various characteristics of residual blocks better.

B. Proposed Rate-Distortion Optimized Transform for Intracoding

To illustrate the distinctive elements of different methods, DCT, MDDT and the proposed RDOT are compared in

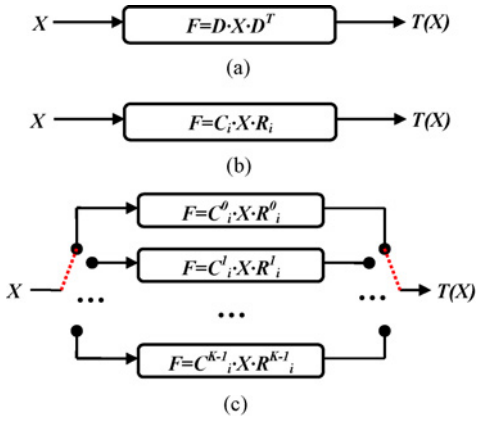

Fig. 4. Transform schemes of a residual block X for intraprediction mode i in (a) DCT, (b) MDDT, and (c) proposed RDOT.

Fig. 4. As it shows, in conventional DCT, for each prediction residual block X, the TBF are discretized sinusoids. However, in MDDT, the TBF are both KLT-based and mode-dependent for intrapredicted residue, and the column transform functions are not orthogonal with the row transform functions in the original design [32]. Both MDDT and the proposed RDOT are KLT-based, however, in the proposed RDOT, there are K pairs of column and row transform matrix candidates, i.e., $C^0{}_{i,\ldots}C^{K-1}{}_i$ and $R^0{}_{i,\ldots}R^{K-1}{}_i$ for an IP mode i. That is, there are totally K different transform paths for each residual block X. The encoder tries all the candidates and selects the optimal path with minimum rate-distortion (R-D) cost value, which is denoted by the red dotted lines in Fig. 4, for the actual coding of residual block X. Compared with MDDT, for intracoding, our proposed RDOT further refines the transform by imposing both mode and data-dependency on the selection of transform functions, and better energy compaction is achieved in the transform domain.

The proposed RDOT requires additional signaling of the transform indexes between encoder and decoder, such that the decoding process can be correctly implemented without any mismatch. Within the framework of RDOT, transform indexes can be either explicitly signaled or implicitly derived. To explicitly signal the transform indexes, new block-level syntax elements need to be defined and encoded into the bitstream. At the decoder, the syntax elements are extracted from the bitstream and decoded, and the corresponding transform matrix is selected to perform the inverse transform. While in an implicit manner, no side information is conveyed into the bitstream, but additional calculations are necessary for both encoder and decoder to implicitly derive the transform indexes using any available reconstructed information. In our simulations, the explicit way is adopted to signal the transform indexes under considerations of encoding and decoding complexity.

The explicit signaling of transform indexes inevitably introduces overhead bits for each block. In our simulations, it is observed that for some smooth image regions and high QP cases, the promotion of transform efficiency with RDOT can be counteracted by coding the overhead. Therefore, the proposed algorithm is implemented as a collaborative contribution of conventional DCT and RDOT, that is, both conventional DCT and proposed RDOT are used as two alternative schemes for coding the current macroblock (MB). And one-bit flag is signaled in the MB header to indicate whether RDOT is used. Therefore, besides three original DCT-based intramodes including I4MB, I8MB, and I16MB, three additional RDOT-based modes including I4MB_RDOT, I8MB_RDOT, and I16MB_RDOT are also defined for the proposed RDOT.

TABLE I
COMPARISONS OF DIFFERENT NUMBER (K) OF TRANSFORM CANDIDATE FOR 4 × 4 TRANSFORM

Sequences 120 Frames, 832×480 (WVGA)	$K=2$		$K=4$ (proposed)		$K=8$	
	BP	BR	BP	BR	BP	BR
BasketballDrill	0.5	−8.8	0.5	−9.9	0.5	−9.8
BQMall	0.5	−8.6	0.6	−10.1	0.6	−9.6
PartyScene	0.6	−7.0	0.7	−8.7	0.7	−8.9
RaceHorses	0.5	−7.3	0.6	−8.0	0.6	−7.8
Flowervase	0.6	−8.5	0.7	−10.2	0.7	−9.8
Keiba	0.5	−8.9	0.6	−10.8	0.5	−9.7
Mobisode2	0.3	−7.1	0.3	−7.6	0.3	−7.2
Average	0.49	−8.0	0.58	−9.3	0.56	−9.0

Although RDOT achieves superior coding performance over DCT for intracoding, the encoding complexity is also increased by the new coding modes and expensive R-D optimized selection of transform matrix. In each of the new RDOT-based MB modes, the encoder performs transform, quantization, entropy coding, dequantization, inverse transform and reconstruction for each transform function candidate, and evaluates the transform function by the R-D cost value. Therefore, the computational complexity is increased drastically for the proposed RDOT-based scheme. The high encoding complexity becomes a major limitation for its applications in practical video codecs, and low-complexity implementations are required. In a later subsection, we will focus on this issue and develop several fast algorithms to collaboratively accelerate the encoding process.

Furthermore, in our simulations, the number of transform function candidates, i.e., K, is empirically set as 4, 16, and 16 for the Intra_4 × 4, Intra_8 × 8, and Intra_16 × 16 intramodes, respectively. To have a brief evaluation on the number of transform candidates, for 4 × 4 transform, we have made an additional experiment using the same training set but different numbers of transform candidates. The results are shown in Table I, and it is seen that the optimal results are obtained when K is set as 4. For coding the side information, the transform index is first binarized as $\log_{2K}$ binary digits. For example, for the case of 8 × 8 transform where K is 16, transform index 3 is binarized as "0011," and 12 is binarized as "1100." Then for coding each binary digit, a single context model using no neighboring auxiliary information is utilized, and the context model is initialized with equal probability.

C. Proposed Rate-Distortion Optimized Transform for Intercoding

For intercoding, the implementation of RDOT is quite similar to the intracase. However, the mode dependency of

the transform candidate is removed since the interprediction residues do not present obvious mode-dependent distribution. Similar as intracoding, besides the original 16 × 16, 16 × 8, 8×16, P8×8 (4×4, 4×8, 8×4, 8×8) MB partition modes, new Inter_16×16_RDOT, Inter_16×8_RDOT, Inter_8 × 16 and P8 × 8_RDOT modes which employ RDOT-based transform are also available. The usage of whether conventional mode or RDOT-based mode is also explicitly signaled with one-bit flag at the MB header, and for case of RDOT-based mode being selected, the transform indexes for each block with nonzero coded block pattern value are also encoded.

The number of candidate transform functions, i.e., K, is empirically set as 2 and 4 for the 4 × 4 and 8 × 8 transforms in intercoding of H.264/AVC, respectively. And for different intercoding modes, the same set of candidate transform matrices are employed. The transform information was signaled into the bitstream after the coded block pattern (CBP) syntax element in each macroblock header, and the transform index of a block will only be coded for a nonzero CBP case. The entropy coding process of the side information is similar as the intracase described previously.

D. Fast Implementation Algorithm of Proposed RDOT

Although the proposed RDOT achieves superior coding performance over the conventional DCT and MDDT, the computational complexity of encoding process is also increased drastically, i.e., about 8–10 times for intracoding compared to H.264/AVC. To break through the bottleneck of high encoding complexity for a practical application, in this subsection, we present the design of an effective fast RDOT method.

The basic idea of the proposed fast implementation is to use the coding results of DCT-based modes for skipping unnecessary RDOT-based modes. Based on the assumption that the optimal DCT and RDOT-based coding modes are highly correlated for both MB-level and block-level coding, we propose both MB-level and block-level R-D cost thresholding techniques to collaboratively accelerate the encoding process by skipping unnecessary RDOT mode trials during the mode decision process.

Before the illustration of proposed methods, let us first introduce a few notations. The R-D cost value of DCT-based mode IxMB is denoted as RDIx, where $x = 4, 8, 16$, and the minimum value of these three cost values is denoted by $RD_{\min}$. All the DCT-based modes are implemented prior to RDOT-based modes, and before checking an RDOT-based mode *IxMB_RDOT*, the following condition is examined:

$$RD_{Ix} < RD_{\min} \times T_{MB} \tag{2}$$

where $x = 4, 8, 16$, and T_{MB} is a user-defined positive constant larger than 1. If the above condition is satisfied, then the corresponding RDOT-based mode *IxMB_RDOT* will be implemented, otherwise, it will be skipped. For example, let us assume that the R-D cost values are 8274, 7700, and 12 182 for I4MB, I8MB, and I16MB, respectively, and therefore $RD_{\min}$ is 7700. For T_{MB} being set as 1.1, the condition 2 will not be satisfied by I16MB, and I16MB_RDOT will be skipped according to the method.

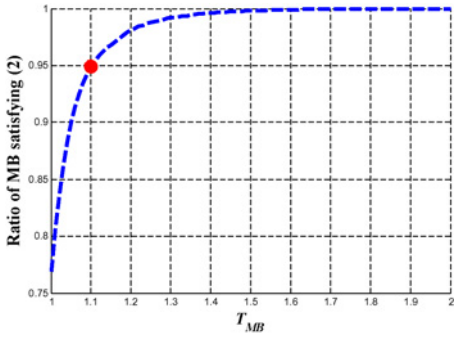

Fig. 5. Ratio of macroblocks where the optimal RDOT-based mode satisfying condition (2) for different TMB threshold values.

For block-level mode skipping, the implementation is similar to the MB-level method indicated above. For an $x \times x$ sub-block in DCT-based MB mode *IxMB*, let us denote the R-D cost value using prediction mode y as $RD_{Dy.}$ The minimum R-D cost value of all prediction directions is recorded as $RD_{\min}$. After that, the following condition is examined for each prediction direction prior to each RDOT-based mode *IxMB_RDOT*:

$$RD_{Dy} < RD_{\min} \times T_{\text{Block}} \tag{3}$$

where $x = 4, 8, 16$, and T_{Block} is also a user-defined constant larger than 1. If the above condition (3) is not satisfied, then the corresponding prediction direction in RDOT-based mode will be skipped, else it will be implemented. For example, let us consider a 16 × 16 block using 16 × 16 prediction, assume that the R-D cost values of prediction Modes 0, 1, 2, and 3 are 2149, 1433, 1355, and 1457, respectively, and $RD_{\min}$ equals 1355. For T_{Block} being set as 1.1, it is noted that except for Mode 0 (horizontal), all the left modes satisfy the above condition, therefore, mode 0 will be excluded from the mode decision.

To reveal the practicability of the above proposed MB-level and block-level thresholding methods, we made simulations on a collection of video content to measure the hit probability of the above proposed conditions (2) and (3). In Fig. 5, the ratio of MB where the optimal RDOT-based modes are successfully included by condition (2) is shown. As it shows, for T_{MB} being set as 1 which allows only one RDOT-based mode being checked, about 77% macroblocks are successfully checked using the optimal RDOT-based mode. This validates the high correlation between DCT-based and RDOT-based coding modes. Also the results for block-level modes are shown in Fig. 6. As it shows, over 95% blocks are successfully checked using the optimal RDOT-based mode when T_{Block} is set as larger than 1.1, 1.3, and 1.5 for Intra16 × 16, Intra8 × 8, and Intra4 × 4 blocks, respectively. Based on the above results, it is verified that the DCT-based and RDOT-based coding modes are highly correlated. With extensive simulations, T_{MB} is set as 1.1 (highlighted by red point in Fig. 5) which allows the optimal RDOT mode of about 95% macroblocks being covered by (2), and T_{Block} is set as 2.0, 1.4, and 1.15 (highlighted by red point in Fig. 6) for I4 MB, I8 MB, and I16 MB, respectively.

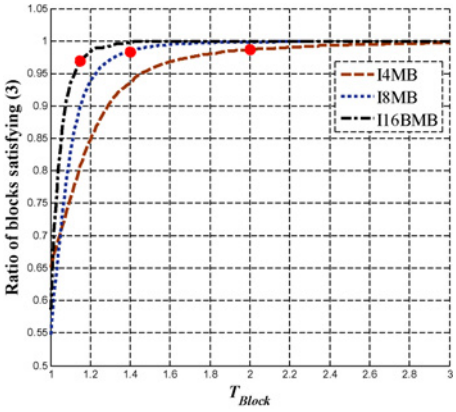

Fig. 6. Ratio of blocks where the optimal RDOT-based mode satisfying condition (2) for different TBlock threshold values.

Additionally, we have incorporated one more improvement which is denoted as luminance coding speedup (LCS). LCS is actually an optimization to the source code with no performance degradation, and was originally designed for accelerating the rate-distortion optimized quantization (RDO-Q) [28] technique in KTA. Generally, LCS stores the luminance coding results (optimal modes, CBP values, coefficients, and so on) when coding mode of chroma is 0 (DC_PRED), and simply restores these results for the remaining chroma coding modes. Therefore, redundant luminance coding in the chroma coding loop can be avoided. In the proposed scheme, we made this method compatible with RDOT when RDO-Q is disabled. To distinguish the contribution of LCS, the results with and without using LCS are both tabulated in Section V.

The proposed fast implementation is simple and does not introduce much additional computations, but significantly accelerates the encoding process with only negligible coding performance degradation.

IV. Analysis and Implementation of the Training Process

Within the framework of our proposed RDOT, there is still one important issue which has not been discussed but evidently affects the coding performance. How to obtain the optimal group of transform matrix candidates? In this section, we will focus on this issue and present both in-depth analysis and detailed technical design on the optimal group of transform matrix candidates.

A. Optimal Transform Matrix Candidates Group

To measure the efficiency of a certain transform, the energy packing efficiency (EPE) criterion is employed [21], [29], [30]. Consider the case of transforming a group of M vectors $\{X_m\}$ with a single transform function $T(\cdot)$, where $m = 0, 1, \ldots, M-1$, and each vector contains N elements. The EPE of $T(\cdot)$ on $\{X_m\}$ is the sum of energy ratio of the first N_0 transform coefficients to the total energy (contained in all N transform coefficients)

$$EPE_{N_0} = \sum_{m=0}^{M-1} \left(\sum_{n=0}^{N_0-1} F_{mn}^2 \Big/ \sum_{n=0}^{N-1} F_{mn}^2 \right) \tag{4}$$

where F_m indicates the mth transform coefficient vector obtained by $F_m = T(X_m)$. For single-transform based schemes, the maximum EPE is achieved by KLT, where the basis functions are the eigenvectors of the covariance matrix of X.

In the scenario of multi-transform based scheme, the transform function for each X_m is selected as the optimum one in a candidate set $\{T_k jk = 0, 1, \ldots, K-1\}$. According to the above principle, the EPE of the candidate set is modified as

$$EPE_{N_0} = \sum_{m=0}^{M-1} \max_{k=0,1,\ldots,K-1} \left\{ \sum_{n=0}^{N_0-1} F_{mn}^2 \Big/ \sum_{n=0}^{N-1} F_{mn}^2 \,|F_m = T_k(X_m) \right\} \tag{5}$$

where T_k indicates one of the transform function candidates. Given the definition of EPE of a transform function candidate set $\{T_{kj}k = 0, \ldots, K-1\}$, the optimal set of transform candidates $\{T_k^*\}$ then corresponds to the solution of the following optimization problem, that is:

$$\max \left\{ \sum_{m=0}^{M-1} \max_{k=0,1,\ldots,K-1} \left\{ \sum_{n=0}^{N_0-1} F_{mn}^2 | F_m = T_k(X_m) \right\} \right\}$$
$$\text{subject to } (x, x) = (T_k(x), T_k(x)), \forall x \in \mathrm{R}^m, \; k = 0, 1, \ldots, K-1 \tag{6}$$

where (x, y) indicates inner product of x and y, and $(x, x) = (T_k(x), T_k(x))$ restricts T_k to be an orthogonal transform. Because T_k are all orthogonal transforms, the following equation holds:

$$\sum_{n=0}^{N-1} F_{mn}^2 = \sum_{n=0}^{N-1} X_{mn}^2 \tag{7}$$

and the denominator term in (5) is omitted in (6) since it is a constant. The cost function in the optimization problem of (6) is non-convex, and more than one local optimum exists in the space spanned by all orthogonal transforms. In the next subsection, an iterative training process will be developed for searching a sub-optimal solution of (6). The global optimum can be approximately obtained in a naive way by running the training process many times and selects the best sub-optimal solution which minimizes the cost function in (6).

B. Proposed Training Process

Within the framework of 2-D separable transform, the transform function is defined by column (C) and row (R) transform matrix. Therefore, the solution of (6) is actually a set of two tuples $\{(C_k{}^*, R_k{}^*) | k = 0, 1, \ldots, K-1\}$. Furthermore, it is enlightening to see that the optimization problem of (6) can be also regarded as a clustering problem, where K tuples (C_k, R_k) are the centers of K clusters, and each training block B_m is assigned to the nearest cluster. Motivated by the k-means clustering X [31] X which is a classical method in cluster analysis, a two-step iterative training method is proposed to obtain a set of column and row transform matrix candidates. The proposed method is an iterative refinement technique composed of an initialization step and two iterative steps.

1) *Initialization*

   a) Randomly label each training block to one of the K clusters.

Input:

1. Training set T, of size $M \times N$, where M/N equals the number of training blocks, N=4, 8 and 16 for I4MB, I8MB and I16MB, respectively.
2. Number of column and row transforms K.

Output:

K candidate column and row transform matrices.

(0) Initialization:

Randomly assign each residual block to one of the K clusters; Calculate the column and row transform matrices using SVD for each cluster, and assign the pair of transform matrices as the cluster center; Set *PreEPE* as -1

(1) Iteration:

Assignment Step: For each residual block, find the nearest cluster center which maximizes the EPE, and assign the residual block to the nearest cluster

Refinement Step: For each cluster, calculate the column and row transform matrices using SVD, and assign the new pair of transform matrices as the cluster center;

Calculate the total EPE as *CurEPE*, if *CurEPE*>*PreEPE*, set *PreEPE* as *CurEPE*, and go to the **Assignement Step**, else exit the iteration and output the K cluster centers.

Fig. 7. Proposed iterative training method for generating the transform matrix candidates.

TABLE II

COMPARISONS OF CODING PERFORMANCE USING DIFFERENT N_0 SETTINGS IN ALL-INTRATEST CONDITION

Test Sequence 120 Frames	$N_0 = 1$		$N_0 = N/4$ (proposed)	
832×480 (WVGA)	BP*	BR*	BP	BR
BasketballDrill	0.5	−8.8	0.5	−9.9
BQMall	0.5	−8.6	0.6	−10.1
PartyScene	0.6	−7.0	0.7	−8.7
RaceHorses	0.5	−7.3	0.6	−8.0
Flowervase	0.6	−8.5	0.7	−10.2
Keiba	0.5	−8.9	0.6	−10.8
Mobisode2	0.3	−7.1	0.3	−7.6
Average	0.49	−8.0	0.58	−9.3

* BP indicates BD-PSNR, BR indicates BD-rate.

b) For each cluster, calculate the optimal column and row transform matrices using singular value decomposition (SVD) as the cluster center.

2) *Iteration*

a) Assignment step: assign each training block to the nearest center which maximizes the EPE.

b) Refinement step: for each cluster, calculate the optimal column and row transform matrices using SVD, and update the cluster center.

The iteration process terminates when the maximum iteration number is achieved or the iteration becomes converged, i.e., transform matrix candidates no longer increase the total EPE in (6) after the refinement step. Then the K cluster centers are output as the final transform matrix candidates. A practical procedure for the actual training process is also illustrated in Fig. 7, and a MATLAB implementation of the training process is available at [35]. At the assignment step shown in Fig. 7, we use the first quarter of coefficients to evaluate the EPE, i.e., $N_0 = N/4$. For the effects of different N_0 settings, we have made an additional simulation between two different settings: 1) $N_0 = 1$, and 2) $N_0 = N/4$ using the same training set. The results are shown in Table II, and it is observed that the latter setting presents some improvements.

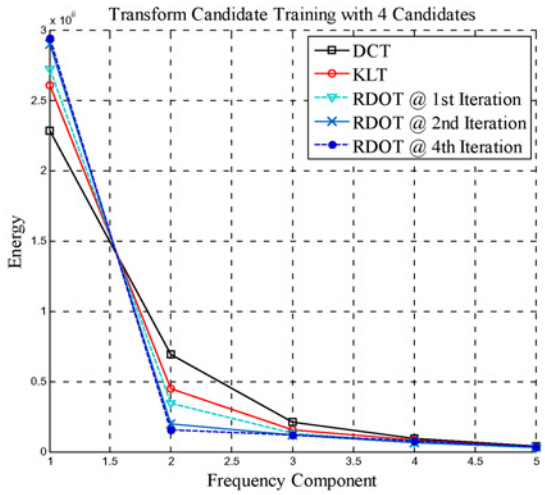

Fig. 8. Energy distributions in transform domain of different transform methods.

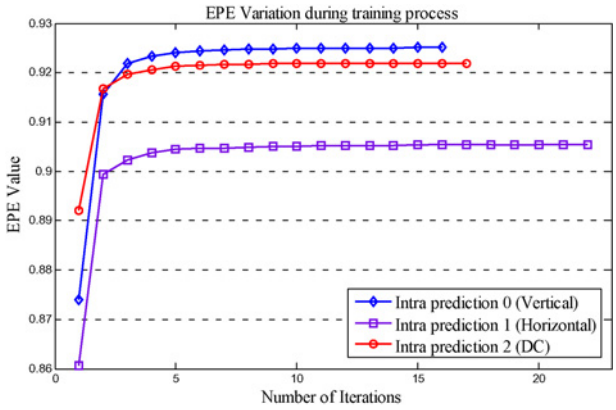

Fig. 9. Variation of EPE value during the training process for intra 8×8 mode 0 (vertical), 1 (horizontal), and 2 (DC).

Iteration of the above training process converges to one local optimum of the cost function in (6), and a brief proof of the convergence is shown in Appendix I. For a subjective evaluation on the changes of the transform efficiency during the iteration process, an actual training process for generating four 8×8 transform candidates is shown in Fig. 8. From Fig. 8 it is noted that the multi-transform-based scheme, i.e., RDOT, significantly outperforms the single-transform based scheme, e.g., KLT and DCT, in terms of energy compaction. It is also observed in Fig. 8 that better energy compaction is achieved in transform domain at each iteration during the training process. The EPE variation during the training process for several IP modes is also shown in Fig. 9 for an illustration of the convergence. The training set is obtained

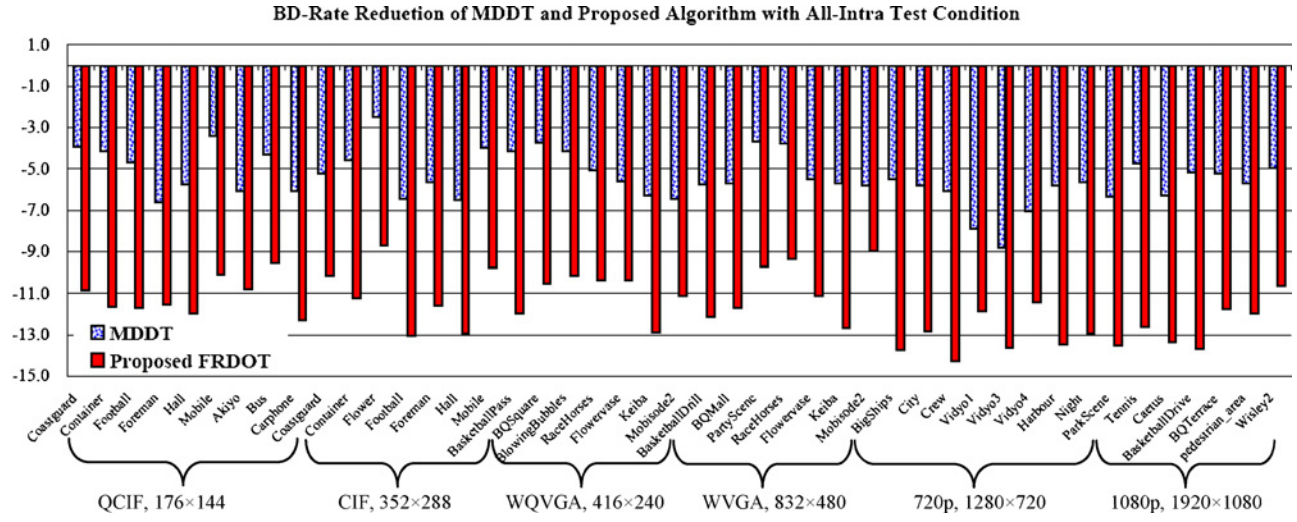

Fig. 10. Coding gains of MDDT and proposed FRDOT under all-intratest condition.

TABLE III
PERCENTAGE OF BLOCKS WITH CODED TRANSFORM INDEX FOR DIFFERENT SEQUENCES AND QP VALUES UNDER ALL-INTRATEST CONDITION

Sequence	QP	Ratio of Blocks with Coded Transform Index		
		Intra_4×4	Intra_8×8	Intra_16×16
Hall (CIF, 352×288)	22	77.8%	85.8%	94.4%
	27	66.1%	70.3%	86.3%
	32	53.8%	56.8%	68.0%
	37	40.5%	42.6%	52.1%
BasketballPass (WQVGA, 416×240)	22	84.9%	93.3%	96.6%
	27	69.0%	81.3%	92.9%
	32	53.5%	60.4%	80.5%
	37	38.1%	45.7%	55.9%
Flowervase (WVGA, 832 × 480)	22	67.9%	29.1%	51.5%
	27	55.3%	26.7%	41.7%
	32	42.7%	17.0%	39.8%
	37	31.8%	11.1%	21.5%
Crew (720p, 1280 × 720)	22	58.6%	82.3%	80.1%
	27	37.6%	64.2%	58.7%
	32	25.1%	36.5%	46.8%
	37	15.2%	16.1%	37.6%

by coding the first frame of several CIF and 720p sequence, and the test set is much larger than the training set. Transform candidates obtained from the above training process for IP mode 0 (vertical), 1 (horizontal), 2 (DC) in I4MB are shown in Appendix II, and a complete set of transform candidates used in experiments is available in [36].

V. EXPERIMENTAL RESULTS

To validate the coding performance of the proposed RDOT, we have integrated the proposed method to the latest KTA software KTA2.6r1 [32]. In this section, three experiments are designed: 1) investigations on the ratios of RDOT-based modes in actual video coding; 2) comparisons between RDOT and different anchors; and 3) comparisons of R-D coding performance and encoding complexity in terms of total encoding time between the original exhaustive RDOT and fast RDOT.

TABLE IV
ENCODING CONFIGURATIONS IN THE EXPERIMENTS

Platforms	Encoding Configurations	
	All intra	IPPP
KTA 2.6r1	High profile, FrameSkip=0, NumberBFrames=0, all KTA tools disabled, RDO-Q off, QP={22, 27, 32, 37}, CABAC, the first 4 s of each test sequence is coded	
	IntraPeriod = 1, all available intraprediction modes including I4MB, I8MB, I16MB are turned on, 4, 16 and 16 candidate transforms are utilized for I4MB, I8MB, and I16MB, respectively	IntraPeriod = 1 s, fast ME, SearchRange = 64, all available MB partition modes including 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4 are turned on, 2 and 4 candidate transforms are utilized for 4×4 and 8×8 transforms, respectively

A. Investigations on the Ratios of RDOT-Based Modes in Actual Video Coding

In this subsection, the utilization of RDOT-based modes in actual video coding is investigated. To explore the characteristics of RDOT-preferred macroblocks, experiments have been conducted on several sequences with different types of video context. As it is shown in Table III, blocks with coded transform index in RDOT-based modes are efficiently used in actual video coding process, especially for high bit-rate cases. The utilization of RDOT-based modes decreases for high QP cases because the contribution of transform is weakened by strong quantization.

B. R-D Performance of MDDT and Proposed Algorithm Compared to H.264/AVC

In this experiment, R-D coding performance of MDDT and proposed RDOT compared to H.264/AVC is shown. To validate the performance of RDOT on various video contexts and resolutions, extensive experiments have been made on a wide range of test set including QCIF (176 × 144), CIF (352 × 288), WQVGA (416 × 240), WVGA (832 × 480),

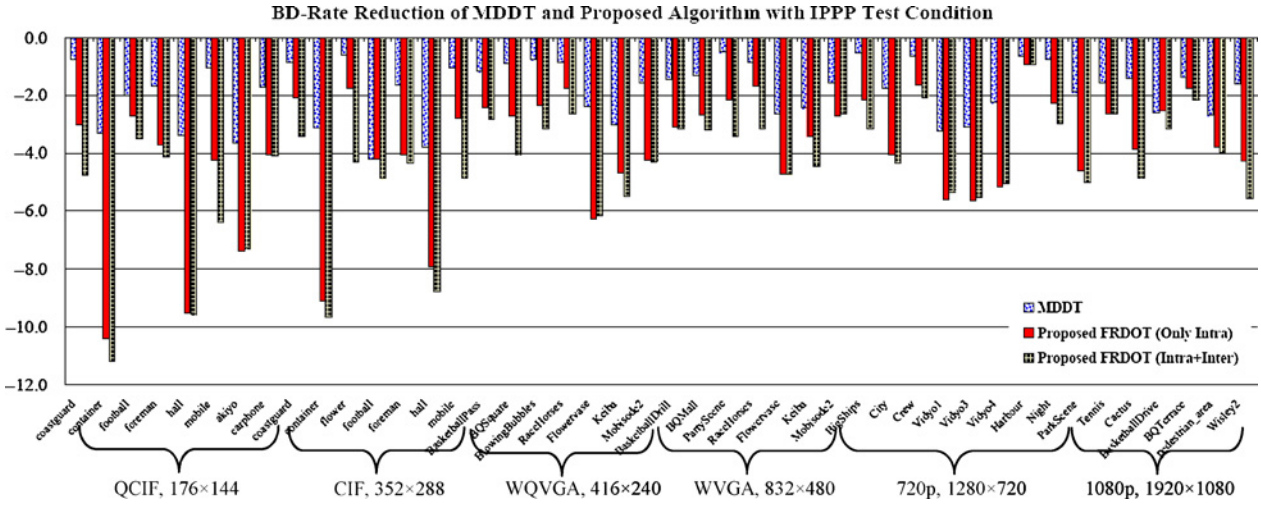

Fig. 11. Coding gains of MDDT and proposed FRDOT under IPPP test condition.

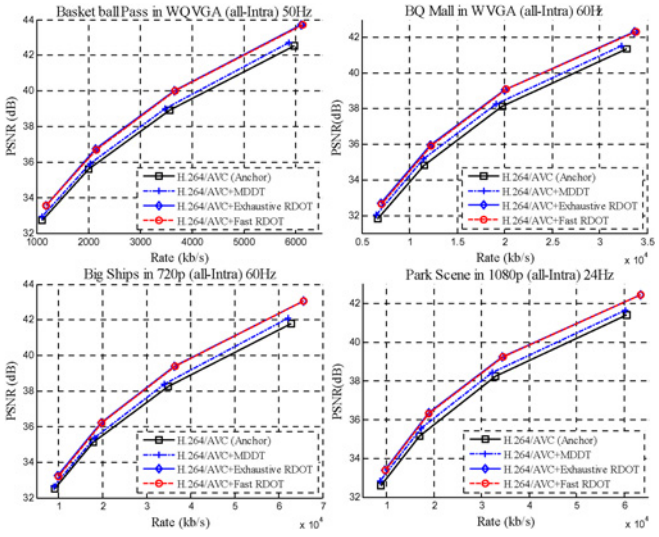

Fig. 12. R-D performance comparisons under all-intratest condition.

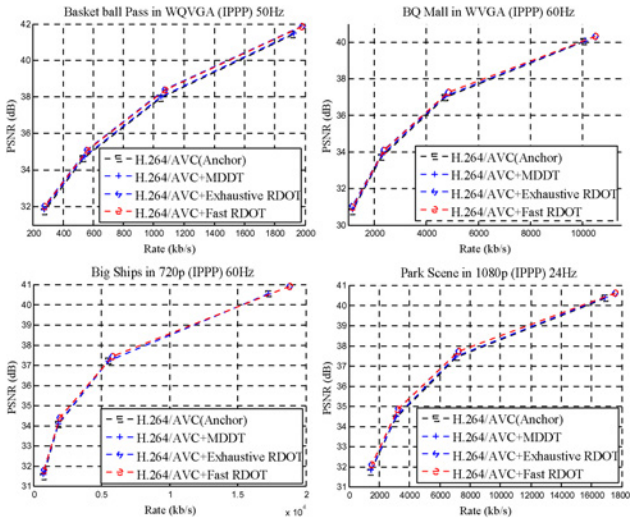

Fig. 13. R-D performance comparisons under IPPP test condition.

720p (1280 × 720), and 1080p (1920 × 1080) formats. These sequences are widely used in research fields related to video coding and the official call for proposals [34] jointly issued by MPEG and VCEG.

Since the proposed RDOT presents different algorithm structures for intracoding and intercoding, i.e, different mode-dependencies, two sets of encoding configurations are employed in our test to better learn the behavior of RDOT: 1) all-intracoding, and 2) IPPP coding. For both configurations, simulations are run on the first 4 s content of each sequence, and the fast RDOT algorithm including LCS has been employed for the proposed method in this experiment. Some important encoding configurations are shown in Table IV. When calculating the average difference between two R-D curves, we employ the popular BD-rate and BD-PSNR [33] for performance evaluations.

The experimental results tabulated in Table V show that, for intracoding, compared to H.264/AVC, MDDT achieves about average 5.5% of BD-rate reduction, or 0.3 dB gain of BD-YPSNR, while the proposed RDOT achieves about average 11.5% of BD-rate reduction and 0.8 dB gain of BD-YPSNR. For intercoding with IPPP condition, about average 1.8% of BD-rate reduction, and 0.1 dB gain of BD-YPSNR is achieved by MDDT, while the proposed RDOT achieves about average 4.6% of BD-rate reduction, and 0.2 dB gain of BD-YPSNR. It is noted that the coding gain in terms of BD-rate reduction achieved is relatively stable for different resolutions. For intracoding, the worst case in terms of BD-rate reduction is 8.7%, while the best case is 14.3%. Also for a grasp of the results shown in Table V, coding gain for intra and intercoding have been shown in Figs. 10 and 11, respectively. The coding gain of RDOT on intercoding is much smaller than the intracase. However, in intercoding where the motion estimation consumes the major encoding time and is separated from transform stage, the additional complexity of RDOT in intercoding is relatively much lower than the intracase. For a comparison of the R-D behavior over the entire QP range, the R-D curves of the anchor, MDDT and proposed methods are shown in Figs. 12 and 13 for

TABLE V
PERCENTAGE OF BLOCKS WITH CODED TRANSFORM INDEX FOR DIFFERENT SEQUENCES UNDER ALL-INTRA AND IPPP TEST CONDITION

Sequence	All-Intra				IPPP					
	MDDT		Proposed FRDOT		MDDT		Intra-FRDOT		Intra and Inter-FRDOT	
	BP*	BR*	BP	BR	BP	BR	BP	BR	BP	BR
Coastguard, 30 Hz	0.3	−3.9	0.8	−10.8	0.0	−0.7	0.1	−3.0	0.2	−4.8
Container, 30 Hz	0.3	−4.1	1.0	−11.7	0.2	−3.3	0.5	−10.4	0.5	−11.2
Football, 30 Hz	0.3	−4.7	0.9	−11.7	0.1	−2.0	0.2	−2.7	0.2	−3.5
Foreman, 30 Hz	0.5	−6.6	0.9	−11.5	0.1	−1.7	0.2	−3.7	0.2	−4.1
Hall, 30 Hz	0.5	−5.8	1.1	−12.0	0.2	−3.4	0.5	−9.5	0.6	−9.6
Mobile, 30 Hz	0.4	−3.4	1.2	−10.1	0.1	−1.0	0.2	−4.2	0.4	−6.4
QCIF Average	0.4	−4.8	1.0	−11.3	0.1	−2.0	0.3	−5.2	0.3	−6.6
Akiyo, 30 Hz	0.4	−6.1	0.8	−10.8	0.2	−3.7	0.3	−7.4	0.3	−7.3
Carphone, 30 Hz	0.4	−6.1	0.8	−12.3	0.1	−1.7	0.2	−4.1	0.2	−4.1
Coastguard, 30 Hz	0.4	−5.2	0.7	−10.2	0.0	−0.8	0.1	−2.1	0.2	−3.4
Container, 30 Hz	0.3	−4.6	0.8	−11.2	0.1	−3.1	0.4	−9.1	0.4	−9.7
Flower, 30 Hz	0.3	−2.5	1.1	−8.7	0.0	−0.6	0.1	−1.8	0.3	−4.3
Football, 30 Hz	0.4	−6.5	0.9	−13.1	0.2	−4.2	0.2	−4.2	0.3	−4.9
Foreman, 30 Hz	0.3	−5.6	0.7	−11.6	0.1	−1.6	0.2	−4.1	0.2	−4.3
Hall, 30 Hz	0.4	−6.5	0.9	−13.0	0.1	−3.8	0.2	−7.9	0.3	−8.8
Mobile, 30 Hz	0.4	−4.0	1.1	−9.8	0.1	−1.1	0.1	−2.8	0.3	−4.9
CIF Average	0.4	−5.1	0.9	−11.0	0.1	−2.3	0.2	−4.8	0.3	−5.7
BasketballPass, 50 Hz	0.3	−4.2	0.8	−12.0	0.1	−1.2	0.1	−2.4	0.1	−2.8
BQSquare, 60 Hz	0.3	−3.7	1.0	−10.5	0.0	−0.9	0.1	−2.7	0.2	−4.0
BlowingBubbles, 50 Hz	0.3	−4.2	0.7	−10.1	0.0	−0.8	0.1	−2.4	0.1	−3.2
RaceHorses, 30 Hz	0.3	−5.1	0.7	−10.4	0.0	−0.9	0.1	−1.7	0.1	−2.6
FlowerVase, 30 Hz	0.4	−5.6	0.8	−10.4	0.1	−2.4	0.3	−6.3	0.3	−6.2
Keiba, 30 Hz	0.4	−6.3	0.9	−12.9	0.2	−3.0	0.3	−4.7	0.3	−5.5
Mobisode2, 30 Hz	0.3	−6.5	0.6	−11.1	0.1	−1.6	0.2	−4.2	0.2	−4.3
WQVGA Average	0.3	−5.1	0.8	−11.1	0.1	−1.5	0.2	−3.5	0.2	−4.1
BasketballDrill, 50 Hz	0.3	−5.8	0.7	−12.2	0.1	−1.4	0.1	−3.1	0.1	−3.2
BQMall, 60 Hz	0.3	−5.7	0.7	−11.7	0.1	−1.3	0.1	−2.7	0.1	−3.2
PartyScene, 50 Hz	0.3	−3.7	0.8	−9.7	0.0	−0.5	0.1	−2.2	0.2	−3.4
RaceHorses, 30 Hz	0.3	−3.8	0.7	−9.3	0.0	−0.9	0.1	−1.7	0.1	−3.2
FlowerVase, 30 Hz	0.4	−5.5	0.8	−11.1	0.1	−2.7	0.2	−4.7	0.2	−4.7
Keiba, 30 Hz	0.3	−5.7	0.7	−12.7	0.1	−2.4	0.1	−3.4	0.2	−4.5
Mobisode2, 30 Hz	0.2	−5.8	0.3	−9.0	0.0	−1.6	0.1	−2.7	0.1	−2.6
WVGA Average	0.3	−5.1	0.7	−10.8	0.1	−1.5	0.1	−2.9	0.1	−3.5
BigShips	0.3	−5.5	0.8	−13.8	0.0	−0.5	0.1	−2.2	0.1	−3.2
City	0.4	−5.8	0.9	−12.8	0.1	−1.8	0.1	−4.1	0.1	−4.4
Crew	0.2	−6.1	0.6	−14.3	0.0	−0.6	0.0	−1.6	0.1	−2.1
Vidyo1	0.4	−7.9	0.7	−11.9	0.1	−3.2	0.2	−5.6	0.2	−5.4
Vidyo3	0.5	−8.8	0.9	−13.6	0.1	−3.1	0.2	−5.7	0.2	−5.5
Vidyo4	0.4	−7.0	0.6	−11.5	0.1	−2.2	0.2	−5.2	0.2	−5.1
Harbor	0.4	−5.8	1.1	−13.5	0.0	−0.6	0.0	−0.9	0.0	−0.9
Night	0.4	−5.7	0.9	−12.9	0.0	−0.7	0.1	−2.3	0.1	−3.0
720p Average	0.4	−6.4	0.8	−13.1	0.1	−1.6	0.1	−3.4	0.1	−3.7
ParkScene	0.3	−6.3	0.7	−13.5	0.1	−1.9	0.2	−4.6	0.2	−5.0
Tennis	0.2	−4.7	0.4	−12.6	0.1	−1.6	0.1	−2.6	0.1	−2.6
Cactus	0.3	−6.3	0.6	−13.4	0.0	−1.4	0.1	−3.9	0.1	−4.8
BasketballDrive	0.2	−5.2	0.5	−13.7	0.1	−2.6	0.1	−2.5	0.1	−3.2
BQTerrace	0.3	−5.2	0.6	−11.8	0.0	−1.4	0.1	−1.7	0.1	−2.2
Pedestrian_area	0.2	−5.7	0.4	−12.0	0.1	−2.7	0.1	−3.8	0.1	−4.0
Wisley2	0.4	−4.9	0.8	−10.6	0.1	−1.6	0.2	−4.3	0.2	−5.6
1080p Average	0.2	−5.7	0.5	−11.7	0.1	−1.9	0.1	−3.4	0.1	−3.9
Total Average	0.3	−5.5	0.8	−11.5	0.1	−1.8	0.2	−3.9	0.2	−4.6

* BP indicates BD-PSNR, BR indicates BD-rate.

all-intra and IPPP test conditions, respectively. From both Figs. 12 and 13 it is seen that, for all the test sequences, our proposed method outperforms the anchor and MDDT over the full range of QP values. Also for an investigation of the characteristics of blocks preferred by RDOT, MB partitions in one frame of *BQSquare* (416 × 240) and *Mobile* (352 × 288) coded as intra at QP = 37 is shown in Fig. 14(a) and (b), respectively, and blocks with coded transform index are highlighted with red. It is clearly shown in Fig. 14 that, blocks with textures, i.e., high-energy prediction residue, are coded with a transform index using RDOT-based mode for most of the time.

C. Comparisons Between the Exhaustive RDOT and Fast RDOT

In this subsection, we conducted several comparisons on the complexity of different algorithms. Both the encoding and decoding complexity are evaluated in this experiment, but we mainly focus on the encoding complexity since it takes the major part of algorithm complexity. The overall

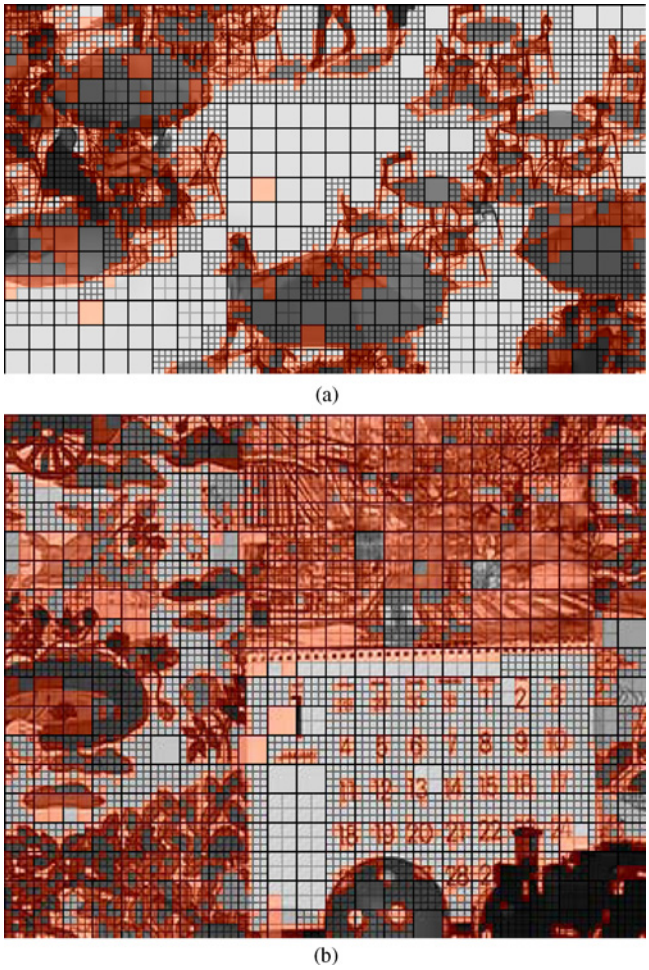

Fig. 14. R-D performance comparisons under all-intratest condition. (a) BQSquare_416 × 240, QP = 37. (b) Mobile_352 × 288, QP = 37.

TABLE VI
COMPARISONS OF OVERALL ENCODING TIME AMONG ANCHOR, MDDT, PROPOSED EXHAUSTIVE RDOT AND PROPOSED FAST RDOT (FRDOT) UNDER ALL-INTRATEST CONDITION

Sequences 120 Frames	QP	Total Encoding Time (in s)			
		Anchor	MDDT	FRDOT*	FRDOT**
BasketballPass WQVGA	22	377	450	2532	730
	27	320	388	1938	562
	32	282	344	1469	431
	37	257	315	1177	349
BQMall WVGA	22	1940	2336	13 020	3651
	27	1653	2004	9575	2702
	32	1441	1772	7331	2102
	37	1313	1610	5797	1682
BigShips 720p	22	4185	5058	33 102	9570
	27	3599	4338	24 730	7122
	32	3079	3865	18 424	5345
	37	2807	3563	13 535	3967
ParkScene 1080p	22	3851	4555	32 536	9078
	27	3244	3893	24 539	6858
	32	2857	3475	18 349	5115
	37	2625	3230	13 150	3739
Average $T_{\text{proposed}}/T_{\text{anchor}}$		1	1.22	6.54	1.86

* indicates LCS off, ** indicates LCS on.

program running time T is used to measure the complexity, and $T_{\text{proposed}}/T_{\text{anchor}}$ is used to evaluate the coding time of proposed method compared with the anchor. All the experiments shown in this subsection are performed on an Intel Core 2 Duo E6600 2.4 GHz personal computer with 3 GB random access memory.

In this experiment, we have two anchors and two proposed methods: 1) anchor 1: KTA2.6r1 with MDDT off; 2) anchor 2: KTA2.6r1 with MDDT on; 3) proposed FRDOT with LCS disabled; and 4) proposed FRDOT with LCS on. The overall CPU time of the four methods on different sequences is shown in Table VI. Comparisons of the R-D performance between the proposed fast RDOT method and the above two anchors have also been shown in Figs. 12 and 13. From Figs. 12 and 13, it is validated that, for all the sequences shown in the figures, the proposed fast RDOT method closely approaches the original exhaustive RDOT over the full range of QP values, and no visible performance degradation is observed.

VI. CONCLUSION AND FUTURE WORK

In the existing video coding standards, transform is implemented with DCT basis functions. However, for some cases, the conventional DCT-based scheme failed to approximate the optimum transform for intraprediction residue. To address this problem, in this paper, we presented the design of a RDOT scheme, where transform is implemented with multiple TBF candidates obtained from off-line training. With the proposed method, the non-stationary characteristic of natural imagery sources is better accommodated than the conventional DCT. Extensive simulations show that, compared to H.264/AVC, the proposed method achieves significant coding gain for a wide range of video resolutions, and superior coding performance is also achieved over the recent MDDT method.

In the future work, further improvements of proposed RDOT will be investigated regarding to the following two issues: 1) lower algorithm complexity for hardware implementation, and 2) further improvement of coding performance from refined algorithm design.

APPENDIX I

In this appendix, a brief proof of the convergence of the training process is given. Let us denote the values of EPE during the training process as a sequence of EPE_{Ai}, EPE_{Ri}, $i = 0, 1, \ldots$, where the subscripts Ai and Ri indicate the assignment step and refinement step at the ith iteration, respectively. To validate the convergence of the training process, we need to prove that the sequence has a limit, that is

$$\lim_{i\to\infty} EPE_{Ai} = \lim_{i\to\infty} EPE_{Ri} = C \tag{8}$$

where C is a constant. First, because the following inequality holds:

$$0 \le \sum_{m=0}^{M-1}\sum_{n=0}^{N_0-1} F_{mn}^2 \le \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F_{mn}^2 \tag{9}$$

the values of the elements in the sequence $\{EPE_{Ai}, EPE_{Ri}, i = 0, 1, \ldots.\}$ are then bounded as

$$0 \leq EPE_{Ai}, EPE_{Ri} \leq 1. \tag{10}$$

Second, at the refinement step, since each cluster center is re-calculated using SVD to minimize the total distance, i.e., maximize the total *EPE*, within the cluster, we have

$$EPE_{Ai} \leq EPE_{Ri}. \tag{11}$$

Furthermore, at the assignment step, each training block is assigned to the nearest cluster of which the cluster center minimizes the *EPE* value, therefore we have

$$EPE_{Ri} \leq EPE_{Ai+1}. \tag{12}$$

According to (10), (11), and (12), it is concluded that EPE_{Ai}, EPE_{Ri}, $i = 0, 1, \ldots.$ is a bounded monotonic sequence, and therefore the sequence has a limit, i.e., the training process is convergent.

APPENDIX II

Transform matrix candidates used in the experiment for IP mode 0 (vertical), 1 (horizontal), 2 (DC) in I4MB are shown below.

IP mode 0 (vertical)

$$C_0 = \begin{bmatrix} 33 & 59 & 75 & 79 \\ 74 & 73 & -11 & -74 \\ -75 & 25 & 79 & -62 \\ 64 & -84 & 66 & -28 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 46 & 66 & 72 & 69 \\ -93 & -46 & 44 & 60 \\ 51 & -55 & -62 & 84 \\ 54 & -83 & 73 & -33 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 37 & 59 & 71 & 80 \\ 38 & 72 & 27 & -95 \\ -93 & -13 & 84 & -22 \\ 70 & -87 & 59 & -21 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 34 & 62 & 76 & 75 \\ 75 & 71 & -19 & -73 \\ -74 & 29 & 75 & -67 \\ 65 & -82 & 68 & -30 \end{bmatrix}$$

$$R_0 = \begin{bmatrix} 7 & -4 & -63 & -111 \\ 28 & 71 & -90 & 50 \\ 67 & 80 & 65 & -35 \\ 105 & -70 & -13 & 17 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 38 & -66 & -92 & 45 \\ 53 & -80 & 44 & -72 \\ 80 & 15 & 60 & 78 \\ 76 & 74 & -48 & -55 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 49 & -108 & 32 & 37 \\ 54 & -25 & -97 & -59 \\ 78 & 57 & -21 & 81 \\ 71 & 31 & 75 & -70 \end{bmatrix}$$

$$R_3 = \begin{bmatrix} -111 & -58 & -24 & -10 \\ -63 & 101 & 44 & 16 \\ -1 & 51 & -118 & -1 \\ 0 & -17 & -9 & 127 \end{bmatrix}.$$

IP mode 1 (horizontal)

$$C_0 = \begin{bmatrix} 49 & 38 & 42 & 104 \\ -24 & -111 & -14 & 57 \\ -105 & 49 & -33 & 44 \\ 50 & 13 & -116 & 18 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 31 & 66 & 82 & 66 \\ -90 & -61 & 40 & 54 \\ 76 & -70 & -29 & 70 \\ 39 & -59 & 85 & -65 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} -7 & 4 & 78 & 101 \\ 10 & 45 & 94 & -74 \\ -88 & -84 & 30 & -26 \\ 92 & -85 & 24 & -9 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} -117 & -52 & -5 & -2 \\ -41 & 87 & 84 & -8 \\ -24 & 65 & -84 & -67 \\ -20 & 45 & -46 & 109 \end{bmatrix}$$

$$R_0 = \begin{bmatrix} 39 & 75 & -77 & 57 \\ 61 & 66 & 32 & -85 \\ 73 & -14 & 76 & 71 \\ 76 & -79 & -60 & -29 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 47 & 89 & -60 & 52 \\ 65 & 50 & 55 & -81 \\ 72 & -40 & 62 & 76 \\ 69 & -66 & -76 & -37 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 31 & 50 & -93 & 65 \\ 59 & 76 & 1 & -85 \\ 75 & 14 & 79 & 66 \\ 79 & -89 & -38 & -25 \end{bmatrix}$$

$$R_3 = \begin{bmatrix} 36 & 62 & -87 & 61 \\ 61 & 71 & 16 & -86 \\ 74 & 0 & 79 & 68 \\ 76 & -86 & -49 & -26 \end{bmatrix}.$$

A complete set of transform candidates for all intraprediction modes of I4/8/16MB is available in [36].

ACKNOWLEDGMENT

The authors would like to thank all the anonymous reviewers for their thoughtful comments and suggestions which helped to improve the technical presentation of this paper.

REFERENCES

[1] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[2] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. 23, no. 1, pp. 90–93, Jan. 1974.

[3] R. J. Clarke, "Relation between the Karhunen–Loève and cosine transforms," *Proc. Inst. Electr. Eng. F*, vol. 128, no. 6, pp. 359–360, Nov. 1981.

[4] K. Karhunen and Kari, "Über lineare methoden in der wahrscheinlichkeitsrechnung," *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.*, no. 37, pp. 1–79, 1947.

[5] M. Loève, *Probability Theory, II* (Graduate Texts in Mathematics, 4th ed.). Berlin, Germany: Springer-Verlag, 1978.

[6] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.

[7] *Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media At Up to About 1.5 Mbit/s: Video*, ISO/IEC 11172-2 (MPEG-1 Video), ISO/IEC, 1993.

[8] *Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Video*, 13818-2-ITU-T Rec. H.262 (MPEG-2 Video), ISO/IEC, 1995.

[9] *Information Technology-Generic Coding of Audio-Visual Objects Part 2: Visual*, ISO/IEC 14496-2 (MPEG-4 Video), ISO/IEC, 1999.

[10] *H.261 Video Codec for Audio Visual Services at $p\times64$ kbits*, Cornit & Consultatif International Telegraphique et Telephonique, Geneva, Switzerland, 1990.

[11] *Video Coding for Low Bitrate Communication*, ITU-T Rec. H.263 Version 1, 1995, Version 2, Sep. 1997.

[12] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec.H.264jISO/IEC 14496-10 AVC)*, document JVT-G050, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Mar. 2003.

[13] AVS Video Expert Group, *Information Technology—Advanced Audio Video Coding Standard Part 2: Video*, document AVS-N1063, Audio Video Coding Standard Group of China (AVS), Dec. 2003.

[14] W.-H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. 25, no. 9, pp. 1004–1009, Sep. 1977.

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

[16] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, Jul. 2003.

[17] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: Theory and design," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 882–893, Apr. 1992.

[18] E. J. Candès and D. L. Donoho, "Ridgelets: A key to higher dimensional intermittency," *Phil. Trans. R. Soc. Lond. A*, vol. 357, no. 1760, pp. 2495–2509, 1999.

[19] J. L. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, Jun. 2002.

[20] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[21] 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.

[22] J. Xu, F. Wu, J. Liang, and W. Zhang, "Directional lapped transforms for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 1, pp. 85–97, Jan. 2010.

[23] 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.

[24] S.-C. Lim, D.-Y. Kim, S. Jeong, J. S. Choi, H. Choi, and Y.-L. Lee, "Rate-distortion optimized adaptive transform coding," *Opt. Eng.*, vol. 48, no. 8, pp. 087004-1–087004-14, Aug. 2009.

[25] X. Zhao, L. Zhang, S. W. Ma, and W. Gao, "Rate-distortion optimized transform for intra-frame coding," in *Proc. IEEE ICASSP*, Mar. 2010, pp. 1414–1416.

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

[27] C. Yeo, Y. H. Tan, Z. Li, and S. Rahardja, *Mode-Dependent Fast Separable KLT for Block-Based Intra Coding*, document JCTVC-B024, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, Jul. 2010.

[28] M. Karczewicz, Y. Ye, and I. Chong, *Rate Distortion Optimized Quantization*, document VCEG-AH21, ITU-T Q.6/SG16 VCEG, Antalya, Turkey, Jan. 2008.

[29] H. Kitajima, "Energy packing efficiency of the Hadamard transform," *IEEE Trans. Commun.*, vol. 24, no. 11, pp. 1256–1258, Nov. 1976.

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

[31] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab. Vol. I: Statist.*, 1967, pp. 281–297.

[32] *Reference ITU-T VCEG-KTA Software*, ITU-T VCEG, version 2.6r1 [Online]. Available: http://iphome.hhi.de/suehring/tml/download/KTA/H

[33] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD Curves*, document VCEG-M33, ITU-T SG16/Q6, 13th VCEG Meeting, Austin, TX, Apr. 2001.

[34] *Joint Call for Proposals on Video Compression Technology*, document VCEG-AM91, ITU-T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11, JCT-VC, Kyoto, Japan, Jan. 2010.

[35] *Training in RDOT* [Online]. Available: http://www.jdl.ac.cn/user/xzhao/RDOTTraining.zip

[36] *Transform Candidates* [Online]. Available: http://www.jdl.ac.cn/user/xzhao/TransCandRDOT.zip

Xin Zhao received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2006. He is currently working toward the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing.

His current research interests include next-generation image and video coding, video processing, and transmission.

Li Zhang received the B.S. degree in computer science from Dalian Maritime University, Dalian, China, in 2003, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2009.

Since 2009, she has held a post-doctorate position with the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, Beijing. She has published over 20 technical articles in related journals and proceedings in the areas of image and video coding, and video processing. Her current research interests include 2-D/3-D image/video coding, video processing, and transmission.

Siwei Ma (S'03) received the B.S. degree from Shandong Normal University, Jinan, China, in 1999, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.

From 2005 to 2007, he held a post-doctorate position with the University of Southern California, Los Angeles. Then, he joined the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, Beijing, where he is currently an Associate Professor. He has published over 70 technical articles in refereed journals and proceedings in the areas of image and video coding, video processing, video streaming, and transmission.

Wen Gao (M'92–SM'05–F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He is currently a Professor of computer science with the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, Beijing, China. Before joining Peking University, he was a Professor of computer science with the Harbin Institute of Technology, Harbin, China, from 1991 to 1995, and a Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. He has published extensively including five books and over 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interfaces, and bioinformatics.

Dr. Gao served or serves on the editorial boards for several journals, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, *EURASIP Journal of Image Communications*, and *Journal of Visual Communication and Image Representation*. He has chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations.