Efficient H.264/AVC Video Coding with Adaptive Transforms

Miaohui Wang, *Student Member, IEEE*, King Ngi Ngan, *Fellow, IEEE*, and Long Xu, *Member, IEEE*

***Abstract*—Transform has been widely used to remove spatial redundancy of prediction residuals in the modern video coding standards. However, since the residual blocks exhibit diverse characteristics in a video sequence, conventional transform methods with fixed transform kernels may result in low efficiency. To tackle this problem, we propose a novel content adaptive transform framework for the H.264/AVC-based video coding. The proposed method utilizes pixel rearrangement to dynamically adjust the transform kernels to adapt to the video content. In addition, unlike the traditional adaptive transforms, the proposed method obtains the transform kernels from the reconstructed block, and hence it consumes only one logic indicator for each transform unit. Moreover, a spiral-scanning method is developed to reorder the transform coefficients for better entropy coding. Experimental results on the Key Technical Area (KTA) platform show that the proposed method can achieve an average bitrate reduction of about 7.95% and 7.0% under all-intra and low-delay configurations, respectively.**

***Index Terms*—Video coding, transform coding, 2D separable Karhunen-Loève transform, H.264/AVC.**

I. Introduction

In A video sequence, the frames contain both temporally and spatially redundant information. The temporal redundancy is usually removed by motion estimation (ME) and motion compensation (MC), while the spatial redundancy is removed by color space conversion [1], predictive coding [2]–[4] or transform coding [5]. The efficiency of transform coding is due to the fact that most of the block energy is compacted into only a few transform coefficients in the frequency domain. In H.264/AVC transform coding, a 2D separable discrete cosine transform (DCT) (or more precisely integer cosine transform (ICT) [6]) is used to decorrelate prediction residuals. In addition, the H.264/AVC adopts multiple transform block sizes to improve the coding performance. For example, in the Key Technical Area (KTA) reference software [7], it supports various transform sizes ranging from 4×4 to 16×16 and its successor High Efficiency Video Coding (HEVC) [8] standard supports up to 32×32. Due to its high efficiency, the DCT based encoders have been well developed, including JPEG, H.26x families [9] and AVS families [10].

Theoretically speaking, it is well known that the DCT is popular due to its energy compaction ability approximating the Karhunen-Loève transform (KLT) under the assumption of a first order stationary Markov process and Gaussian distributed source [11]. In practice, the DCT has been considered as the most suitable transform to replace the KLT for image compression. In image coding, the separable 2D-DCT achieves promising performance because of high correlations among neighboring pixels in a natural image. On the other hand, the correlations of prediction residual blocks in the H.264/AVC may not be high as the natural images. As a result, the DCT would be unlikely to be the best choice for transform coding when applied to the low correlation residuals [12], [13]. This belief motivates many transforms that mainly fall into two categories: fixed transforms and adaptive transforms.

Fixed transform methods are widely studied to replace the DCT by the alternative transforms. The alternative transform candidates can be 1D-DCT, discrete sine transform (DST) or other predefined transform kernels. The advantages of the fixed transform methods include existing fast algorithms and saving of the overhead bitrate. Zeng *et al.* [14] proposed the directional DCT (D-DCT) for image coding, in which the 1D-DCT is performed along a certain pattern in a block. Chang *et al.* [15] developed the directional adaptive block transform (DA-BT) for image coding with two key differences comparing with the D-DCT method: one is the ordering of the transform coefficients and the other is the choice of the transform directions. Ye *et al.* [16] proposed the mode-dependent directional transform (MDDT) for the H.264/AVC all-intra coding. In [17]–[19], sinusoidal transforms were proposed to decorrelate the H.264/AVC residuals based on the observation that the DST outperforms the DCT when the correlations are very low [5].

Adaptive transform methods have also received growing attention to replace the DCT by the adaptive transform kernels. The advantage of the adaptive transforms is that it provides a better energy compaction than the fixed transforms, but the transform kernels are required to be transmitted to the decoder. Due to the transmission of the transform kernels, the adaptive transform methods result in overhead problem. In our previous works [12] and [13], adaptive transform methods are proposed for residual coding, in which the transform kernels are computed from the decoded block. In [20], a hybrid singular value decomposition DCT (HSVD-DCT) method was proposed for still image compression, where only a few big eigenvalues and

Manuscript received July 25, 2013; revised October 28, 2013 and December 18, 2013; accepted December 20, 2013. Date of publication February 10, 2014; date of current version May 13, 2014. This work was supported in part by a grant from the Research Grants Council of the Hong Kong SAR, China (Project CUHK 416010). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ebroul Izquierdo.

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

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

Digital Object Identifier 10.1109/TMM.2014.2305579

corresponding eigenvectors are transmitted. It should be noted that vector quantization (VQ) is used to encode these eigenvectors. Lin *et al.* [21] proposed a 2D-SVD [22] based method for group of intra frames (GOP) coding. The transform kernels are obtained by the 2D-SVD method based on the GOP (GOP ≥ 8) that are being encoded. Then, these transform kernels are transmitted to the decoder and are used to transform the blocks in the GOP only. Biswas *et al.* [23] proposed to transform the MC residual blocks by the KLT. To save the overhead bitrate, the transform kernels are computed by shifting and rotating the MC block.

In this paper, we propose a content adaptive transform (CAT) framework for the H.264/AVC-based video coding. Unlike the traditional adaptive transform methods, the proposed method derives the adaptive transform kernels from the reconstructed block without causing overhead problem, since the decoder can computed them in the same way. In addition, based on the study of the transform coefficient distributions, we propose a spiral-scanning method to reorder the quantized coefficients before run-length-coding (RLC) and entropy coding. Extensive experimental results show that the proposed method is able to achieve higher PSNR gain than state-of-the-art methods in the H.264/AVC-based video coding.

The rest of the paper is organized as follows. We briefly introduce the related works in Section II. The proposed method is then presented in Section III. The efficiency of the proposed method is analyzed in Section IV. The simulation results are presented in Section V and concluding remarks are given in Section VI.

II. Related Works in the H.264/AVC

In this section, we review three classes of state-of-the-art transform methods in detail: (1) directional transform methods, (2) adaptive DCT/DST methods and (3) secondary transform methods. It should be pointed out that all of them is proposed to improve the H.264/AVC coding performance.

A. Directional Transform Methods

In H.264/AVC, directional intra predictions have been used to remove the spatial redundancies. As shown in Fig. 1, the residual samples are collected in a H.264/AVC 8×8 intra-coding experiment, where the black pixel indicates a small prediction error and the white one indicates a large error. The prediction errors in columns or rows will produce directional edges. For example, in Mode 0 and Mode 1, it shows some horizontal and vertical edges, respectively. For Mode 3, diagonal-down right edges can be easily observed, and similarly for other modes. It is evident that the prediction residuals exhibit different statistical characteristics resulting from different prediction modes. Based on this observation, Ye *et al.* [16] proposed a mode-dependent directional transform (MDDT) method for intra prediction residuals coding. It is known that the directional transform is derived from the non-separable KLT. However, the non-separable KLT entails higher complexity and larger storage space. Therefore, the non-separable KLT is replaced by the separable directional transforms [16]:

$$C_i = U_{v,i}^T X U_{h,i},\ (i = 0, 1, 3, 4, 5, 6, 7, 8) \tag{1}$$

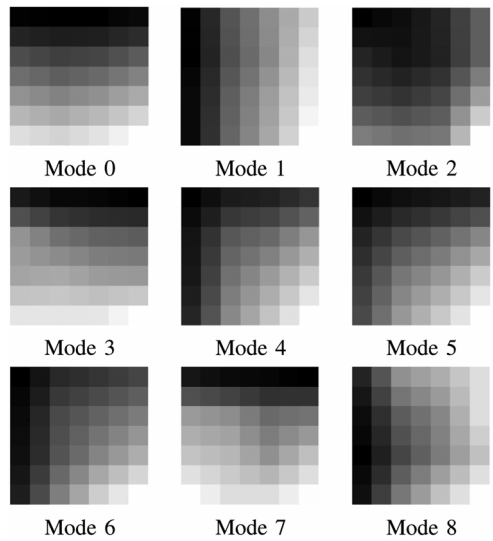

Fig. 1. Normalized magnitudes of the directional prediction residuals in the 8×8 block intra-coding.

where C_i denotes the transform coefficients in mode i. X is the input intra residual block. $U_{v,i}$ and $U_{h,i}$ are the transform basis trained off-line on a large prediction data set. Detail of the training process can be referred to [24]. Note that the DCT is used when the prediction mode is DC (i.e., $i = 2$).

There are a lot of efforts spent in the directional transforms during the past years. Cohen *et al.* [25] proposed a direction adaptive residual transform method in which several 1D transforms were employed along each direction and only the DC coefficients were further processed by the DCT. Sezer *et al.* [26] used ℓ_0-norm regularized optimization method as a more robust way to learn the 2D separable sparse transforms for the directional residual coding. As mentioned above, the MDDT method took only one specific transform basis for each prediction mode, so it might not give full play to its ability for intra residual coding. From this perspective, Zhao *et al.* [24] extended one basis to multiple bases. To choose the best transform basis, the rate distortion optimization (RDO) tool was used. For example, the RDO based transform (RDOT) [24] method used 16 kernels for an 8×8 block in each prediction mode.

B. Adaptive DCT/DST Methods

Based on the analysis of the intra prediction residuals, DST type-VII [18], [19] has been proposed to encode the directional intra residuals. Yeo *et al.* [19] proposed a mode-dependent fast separable KLT (MDFT) method that has similar coding performance as compared with the MDDT method but requires no off-line training. In the MDFT method, the statistics of intra prediction residuals were studied and mode-dependent separable KLT transform kernels were employed for the all-intra coding:

$$U_{klt,i,j} = \frac{2}{\sqrt{2N+1}} \sin\left(\frac{(2i-1)\, j\pi}{2N+1}\right). \tag{2}$$

Based on the covariance models for various prediction residuals, Yeo *et al.* [19] also proposed to use the traditional DCT basis for

TABLE I
CHOICE OF THE KLT/DCT IN THE MDFT METHOD

Mode	Column basis	Row basis
0, 3, 7	U_{klt}	U_{dct}
1, 8	U_{dct}	U_{klt}
2	U_{dct}	U_{dct}
4, 5, 6	U_{klt}	U_{klt}

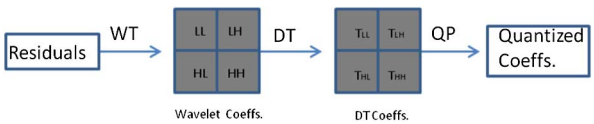

Fig. 2. Flowchart of the two-layer directional transform method.

some prediction modes instead of the MDFT basis (2). The DCT basis is formulated as:

$$U_{dct,i,j} = \frac{w_j}{\sqrt{N}} \cos\left(\frac{(2i+1)\,j\pi}{2N}\right), w_j = \begin{cases} 1 & j = 0 \\ \sqrt{2} & j \neq 0 \end{cases}. \quad (3)$$

The transform kernels of the MDFT method are summarized in Table I.

Adaptive DCT/DST methods have been widely studied. Lim *et al.* [17] explored the possibilities of using DCT/DST type-II to decorrelate the MC residuals. Independently, Han *et al.* [18] analytically derived the DST type-VII along the prediction direction and proposed an asymmetric DST (ADST) method for all-intra coding. They showed that both the upper and the left boundaries of the block were available for prediction for the DC mode, and hence the ADST was used. Otherwise, adaptive ADST/DCT kernels were used for the other modes. Saxena *et al.* [27] proposed an improvement of this work that finally was adopted in recent HEVC standard [8].

C. Secondary Transform Methods

Secondary transform methods have been developed to improve the conventional DCT. Alshina *et al.* [28] used the rotational transforms (ROT) as the secondary transform. The ROT method provided four sets of 4×4 and 8×8 predefined transform kernels for intra residuals. Dong *et al.* [29] proposed a two-layer directional transform method for the MC residual coding. As shown in Fig. 2, the Haar-based discrete wavelet transform (DWT) is employed to quickly compact the block energy to the LL subband in the first layer. In the second layer, 2D non-separable directional transforms (DT) are used, where the transform kernels are off-line trained along the four directions $\{0°, 45°, 90°, 135°\}$.

III. PROPOSED ADAPTIVE TRANSFORM FRAMEWORK FOR VIDEO CODING

This section presents the mathematical foundations behind the proposed method. Firstly, we introduce the content dependent 2D separable Karhunen-Loève Transform (KLT). Secondly, based on the flowchart of the H.264/AVC intra coding, we demonstrate the derivation the content adaptive transform (CAT) framework. Without loss of generality, the CAT model can also be derived in parallel for motion compensation residual coding.

A. 2D Separable Karhunen-Loève Transform (KLT)

KLT is a data dependent transform, and it has been considered as the optimum energy compacting transform if only one fixed transform basis is used for all blocks in an image. When the input and output block are converted to column vectors $\mathbf{x} = [x_1, x_2, \ldots, x_{N^2}]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_{N^2}]^T$ respectively, 1D-KLT is well known as a non-separable transform.

$$\mathbf{y} = T_{klt}\mathbf{x}, \quad (4)$$

where T_{klt} is the transform basis of the 1D-KLT. In practice, we calculate the eigenvalue decomposition of the covariance matrix of the input $\mathbf{x}$ to obtain the transform basis T_{klt}. If $R_{\mathbf{xx}}$ and $R_{\mathbf{yy}}$ denote the covariance matrices of the input vector $\mathbf{x}$ and output vector $\mathbf{y}$ respectively, then we have

$$R_{\mathbf{yy}} = \mathrm{E}\left\{(T_{klt}\mathbf{x}) \cdot (T_{klt}\mathbf{x})^T\right\} = T_{klt} R_{\mathbf{xx}} T_{klt}^T \quad (5)$$

where $\mathrm{E}\{\cdot\}$ is the expectation operator.

The 1D-KLT is easy to extend to the 2D case. Suppose that an image consists of M non-overlapping blocks. The separable 2D-KLT is defined same as (1), $C_{N\times N} = U_{v,N\times N}^T X_{N\times N} U_{h,N\times N}$, and the only difference is the transform kernels. For example, the vertical transform kernel $U_{v,N\times N}$ is given by:

$$U_{v,N\times N} = \begin{bmatrix} | & | & | & | \\ u_{v,1} & u_{v,2} & \ldots & u_{v,N} \\ | & | & | & | \end{bmatrix}, \quad (6)$$

where $u_{v,i}$ is computed as the eigenvector of the vertical covariance matrix $R_{\mathbf{x},\mathbf{v}}$:

$$R_{\mathbf{x},\mathbf{v}} = \frac{1}{M}\sum_{j=1}^{M}\left(X_j - \bar{X}\right)\left(X_j - \bar{X}\right)^T = \frac{1}{M}\bar{X}_v\bar{X}_v^T, \quad (7)$$

where $\bar{X}$ is a column mean vector of $X_j, (j = 1, \ldots, M)$, and $\bar{X}_v = [X_1 - \bar{X}, X_2 - \bar{X}, \ldots, X_M - \bar{X}]$. Similarly, the transform kernel $U_{h,N\times N}$ is computed as the eigenvector of the horizontal covariance matrix $R_{\mathbf{x},\mathbf{h}} = \frac{1}{M}\bar{X}_h^T\bar{X}_h$ and $\bar{X}_h = [(X_1 - \bar{X})^T, (X_2 - \bar{X})^T, \ldots]^T$. Moreover, it can be derived that when the 2D separable KLT is defined for a zero-mean image and computed over only one block X, the horizontal and vertical covariance matrices are estimated as:

$$R_{\mathbf{x},\mathbf{v}} = XX^T, R_{\mathbf{x},\mathbf{h}} = X^TX, \quad (8)$$

and in this case, the 2D separable KLT degenerates into the SVD transform.

From (7), we can conclude that the 2D separable KLT is an optimal transform for an image but not for a specific block. However, if the transform kernels satisfy (8), it is an optimal transform for a block. In the next section, we will introduce the proposed content adaptive transform (CAT) framework that is based on (7) and (8). The proposed transform kernels of the 2D separable KLT can be calculated by method [30].

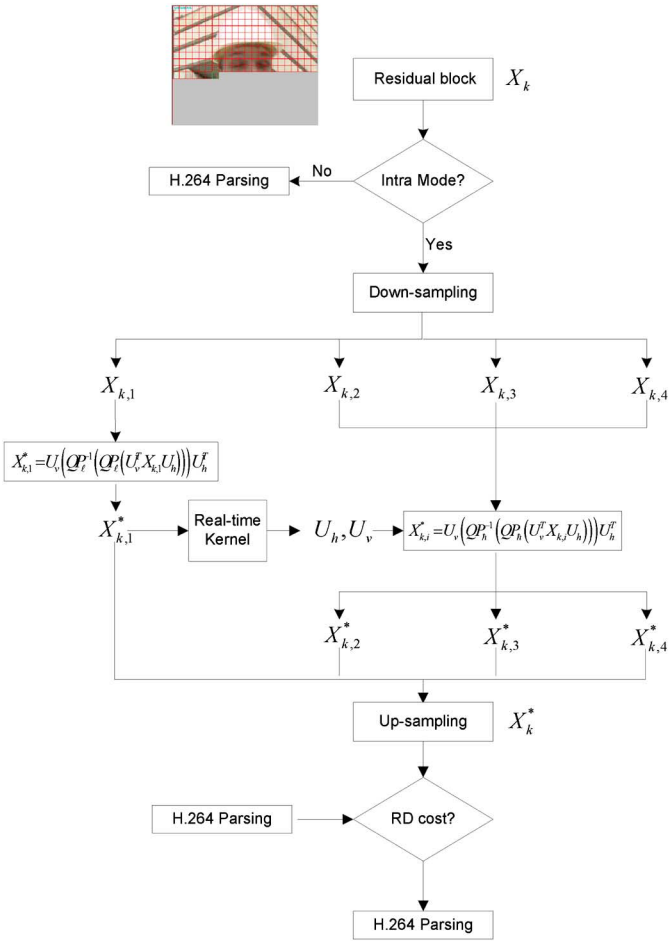

Fig. 3. Illustration of the proposed content adaptive transform framework.

B. Proposed Content Adaptive Transform Framework (CAT) for Video Coding

In video coding, a frame is first divided into non-overlapping blocks for predictive coding, and then the transform coding is performed on each residual block before quantization and entropy coding. To improve the coding performance, the H.264/AVC employs a quad-tree tool to segment the basic coding unit (e.g., macroblock 16×16) based on the rate distortion optimization (RDO). In addition, the H.264/AVC also employs directional prediction to find the best residual block. After the directional prediction, the DCT is employed to transform the intra residuals. Assuming that the kth block $X_{k,N\times N}$ can be transformed as

$$C_{k,N\times N} = U^T_{v,N\times N} X_{k,N\times N} U_{h,N\times N}, \tag{9}$$

where $C_{k,N\times N}$ is the transform coefficients; U_h and U_v are the horizontal and vertical DCT transform kernels, respectively.

As mentioned above, the training kernels of directional transforms depend on the off-line training methods and the quality of training data set. Moreover, the training data contains thousands of directional residual blocks, some of which may be contaminated by noise. The off-line trained kernels are then used to transform the time-varying residuals. As a result, the off-line trained kernels may not adapt well to the content of the time-varying residual blocks. Intuitively, adaptive transform kernels may provide a better coding performance than the off-line trained ones, since they can be better adapted to the content of the residual blocks. To obtain the high quality transform kernels, we propose a content adaptive transform (CAT) framework for video coding. As shown in Fig. 3, the CAT framework consists of two crucial components: sampling unit and transform unit. Each part of the CAT framework is discussed in more detail as follows.

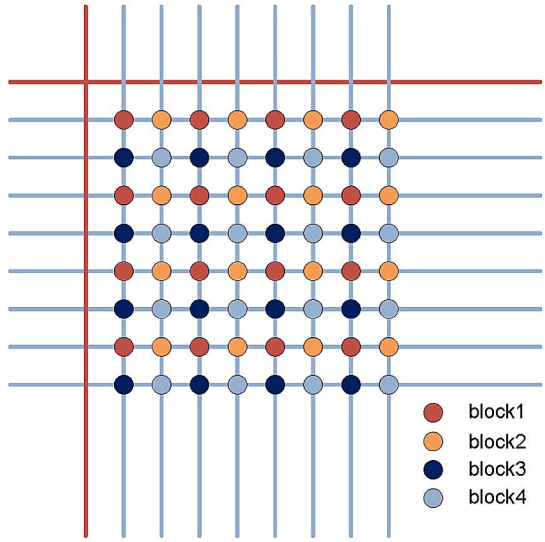

Fig. 4. Illustration of the sampling operator.

In the sampling unit, the down-sampling operator as a pre-processing unit is used to rearrange the residual block X_k into four subsamples, $X_{k,i}(i = 1, 2, 3, 4)$. Correspondingly, the up-sampling operator as a post-processing unit is used to reconstruct the original block X_k from $X_{k,i}(i = 1, 2, 3, 4)$. In the CAT framework, the sampling process involves only shift operations and thus the computational cost is low. The proposed sampling operator is illustrated in Fig. 4. In the transform unit, the block $X_{k,1}$ is first forward-transformed, quantized, dequantized and inverse-transformed as:

$$X^*_{k,1} = U_v \left(QP^{-1}_{\ell,N\times N} \odot \left(QP_{\ell,N\times N} \odot \left(U_v^T X_{k,1} U_h \right) \right) \right) U_h^T \tag{10}$$

where U_h and U_v are the transform kernels of the DCT; $QP_{\ell,N\times N}$ is the quantization method in the H.264/AVC; $\odot$ indicates that each element of $(U_v^T X_{k,1} U_h)_{i,j}$ is processed by H.264/AVC quantization tool, including scaling and rounding operations.

Based on the block $X^*_{k,1}$, the separable 2D-KLT is then used to derive the adaptive transform kernels U_v and U_h. For example, as shown in top-left corner of Fig. 3, the current input block is the 227th macroblock in the "Foreman (CIF)" sequence, and the residual block X_k is predicted by intra direction $MODE = 8$. Transform kernels $U_v = [u_{v,1}, u_{v,2}, \ldots, u_{v,8}]$ and $U_h = [u_{h,1}, u_{h,2}, \ldots, u_{h,8}]$ will be updated as shown in (11) and (12) at the bottom of the next page, where the integers in the matrix are the approximated coefficients of the integer transform, and the scaling factor $\alpha \approx 0.0055$ is integrated into the quantization scheme in the KTA platform. In this

case, the proposed adaptive transform avoids the floating-point arithmetic computation, and hence will not result in the drift problem [6].

After obtaining the real-time transform kernels U_v and U_h, the other blocks will be transformed as:

$$C_{k,i,N\times N} = \sum_{x=1}^{N}\sum_{y=1}^{N} u_{v,x}^{T} X_{k,i}(x,y) u_{h,y}, \tag{13}$$

and then the transform coefficient block $C_{k,i,N\times N}$ is quantized for entropy coding.

For the quantization, different QP values are designed for the two types of blocks, where the $X_{k,1}$ is considered as an important block while the other blocks $\{X_{k,2}, X_{k,3}, X_{k,4}\}$ are considered as ordinary blocks. The reason is that down-sampling may result in low correlations and require more bitrate to reduce the quantization distortion. As a result, to obtain a high quality $X_{k,1}^{*}$, a low QP_{ℓ} should be set for the block $X_{k,1}$. In order to maintain a macroblock level rate distortion cost [31], a high $QP_{\hbar}$ is set for the blocks $\{X_{k,1}, X_{k,2}, X_{k,3}\}$. Based on Monte Carlo simulations, we select the following simple but efficient QP scheme:

$$\begin{cases} QP_{\hbar} = QP \\ QP_{\ell} = QP_{\hbar} - 1 \end{cases} \tag{14}$$

where QP is the macroblock quantization parameter.

Finally, the other blocks $\{X_{k,2}, X_{k,3}, X_{k,4}\}$ are reconstructed by:

$$X_{k,i}^{*} = U_v \left(QP_{\hbar,N\times N}^{-1} \odot (QP_{\hbar,N\times N} \odot C_{k,i,N\times N}) \right) U_h^{T}. \tag{15}$$

The detail of the proposed CAT method is presented in **Algorithm 1**. In the decoder, the important block $X_{k,1}^{*}$ is first reconstructed. Then, the other blocks can be reconstructed by repeating the inverse step (13).

Algorithm 1. The Proposed CAT Algorithm

Input: The input block X_k and frame level QP.

Output: The quantized transform coefficient block $C_{k,i}(i = 1, 2, 3, 4)$.

Begin

1. The traditional DCT part.

1.1 Compute the coefficients of DCT:

$$C_k = U_v^{T} X_k U_h,$$

1.2 Apply QP and QP^{-1} for C_k to get X_k^{*};

1.3 Collect the RD cost. Note that zigzag pattern is used as scanning table.

$$RDCost_{DCT} = Distortion(X_k, X_k^{*}) + \lambda \times Rate(X_k, X_k^{*}).$$

2. The proposed CAT part.

2.1 Apply a downsampling operator to the block X_k;

$$X_k = \{X_{k,1}, X_{k,2}, X_{k,3}, X_{k,4}\}$$

2.2 Compute the coefficients of DCT:

$$C_{k,1} = U_{v,dct}^{T} X_{k,1} U_{h,dct},$$

$$U_{v,8\times 8} \approx \alpha \begin{bmatrix} -44 & 43 & -12 & 9 & -99 & -4 & 49 & -1 \\ -47 & -4 & -58 & 59 & -5 & -12 & -78 & -29 \\ -66 & 36 & -15 & -78 & 29 & -28 & -23 & 43 \\ -60 & -60 & -20 & 33 & 42 & -12 & 73 & 15 \\ -55 & 12 & 78 & -5 & 18 & 15 & -2 & -80 \\ -27 & -76 & 22 & -28 & -48 & 61 & -35 & 31 \\ -2 & -17 & 68 & 33 & -21 & -85 & -24 & 42 \\ 17 & -55 & -29 & -58 & -27 & -62 & 5 & -63 \end{bmatrix} \tag{11}$$

$$U_{h,8\times 8} \approx \alpha \begin{bmatrix} -52 & -58 & -57 & -18 & 55 & 4 & -53 & 23 \\ -43 & -56 & 30 & -22 & 4 & 47 & 86 & 0 \\ -21 & -51 & 16 & 47 & -22 & 20 & -24 & -96 \\ -4 & -45 & 59 & 25 & -54 & -5 & -41 & 73 \\ 26 & -39 & -35 & -77 & -58 & -56 & 14 & -6 \\ 41 & -36 & -47 & 76 & 16 & -35 & 52 & 30 \\ 62 & -33 & 63 & -29 & 75 & -26 & -9 & -9 \\ 70 & -30 & -28 & -1 & -20 & 91 & -24 & -3 \end{bmatrix}. \tag{12}$$

2.3 Apply QP_ℓ, ${QP_\ell}^{-1}$ and inverse transform for $C_{k,1}$ to get $X^*_{k,1}$;

2.4 Collect the RD cost. Note that the spiral method is used.

$$RDCost^1_{CAT} = Distortion(X_{k,1}, X^*_{k,1}) + \lambda \times Rate(X_{k,1}, X^*_{k,1}),$$

2.5 Compute the 2D separable KLT transform kernels:

$$C^*_{k,1} = U_v^T X^*_{k,1} U_h,$$

2.6 Call proposed transform coding.

For $i = 2:4$

$$C_{k,i} = U_v^T X_{k,i} U_h,$$

Apply $QP_\hbar$, ${QP_\hbar}^{-1}$ and the inverse transform for $C_{k,i}$ to get $X^*_{k,i}$;

Collect the RD cost:

$$RDCost^i_{CAT} = Distortion(X_{k,i}, X^*_{k,i}) + \lambda \times Rate(X_{k,i}, X^*_{k,i}),$$

End For

2.7 Compute the overall RD cost:

$$RDCost_{CAT} = \sum_{i=1}^{4} RDCost^i_{CAT}.$$

3. The RDO part.

If $RDCost_{CAT} < RDCost_{DCT}$

$\text{CATMode} = 1;$

else

$\text{CATMode} = 0;$

End If

End

IV. Analysis of the CAT Framework

This section presents the experimental analysis of the CAT framework from various aspects. First, we study the correlations among the subsamples. Second, we discuss the proposed spiral-scanning method for better reordering of the transform coefficients. Third, experiments are carried out to investigate the performance of a transform size and transform coding gain. Finally, we discuss the difference between the proposed method and the training-based directional transform methods.

A. Correlation Analysis

In this subsection, we analyze the correlations among the subsamples from various block sizes in both the original frame and its intra prediction residual. As shown in Fig. 5, "BQMall" and its intra-coding residuals are used in this experiment. Correlation among the subsampled blocks $\{X_{k,1}, X_{k,2}, X_{k,3}, X_{k,4}\}$ is very crucial to the proposed CAT method, which determines the energy compaction efficiency of the adaptive transform kernels. Thus, Pearson correlation coefficient (PCC) is used to measure the strength and the direction of a linear relationship between two subsampled blocks:

$$PCC = \frac{\sum_{i=1}^{n} (Y_i - \bar{Y})(Z_i - \bar{Z})}{\sqrt{\sum_{i=1}^{n} (Y_i - \bar{Y})^2} \sqrt{\sum_{i=1}^{n} (Z_i - \bar{Z})^2}}, \quad (16)$$

where Y_i and Z_i are paired samples. In the statistical perspective, a correlation greater than 0.8 is generally considered to be strong, whereas a correlation less than 0.5 is generally described as weak.

Fig. 6 shows the PCCs among the 4×4 blocks extracted from an 8×8 block by pixel rearrangement (Fig. 4(a)) for both the original "BQMall" frame and its intra prediction residual as shown in Figs. 5(a) and 5(b) ($\text{QP} = 22$), respectively. From Fig. 6, it is easily observed that most of the blocks are on the upper side of the plane with the $\text{PCC} = 0.8$, which indicates that the high correlations indeed exist among the subsampled pixels in an 8×8 block. In addition, the original "BQMall" frame has stronger correlation compared to its intra prediction residual. However, the prediction residual blocks with strong correlations account for 66% of the total. It indicates that for those subsampled blocks with high correlations, the proposed separable 2D-KLT kernels can achieve better energy compaction as formulated by (13).

Fig. 7 further shows the PCCs among subsampled blocks of 16×16 and 32×32 blocks, respectively. It can be seen that the

(a) (b)

Fig. 5. The first frame of "BQMall" (832×480) sequence and its intra prediction residual. The grey color indicates zero residual, the black indicates negative residual, and the white indicates positive residual. (a) "BQMall"original frame. (b) "BQMall" intra residue.

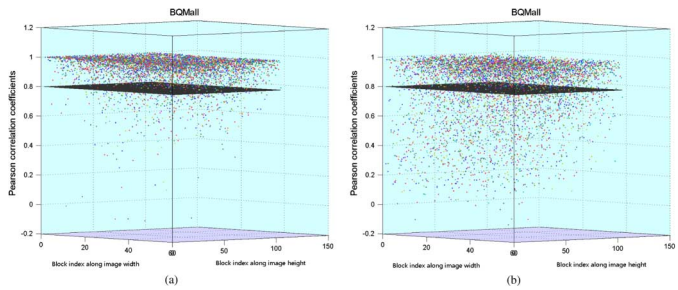

Fig. 6. Plot of the Pearson correlations among subsamples of an 8×8 block (Every 8×8 block is downsampled and correlations are obtained among its subsamples.) (a) Original 8×8 blocks, $(\text{PCC} \geq 0.8) = 87\%$. (b) Residual 8×8 blocks, $(\text{PCC} \geq 0.8) = 66\%$.

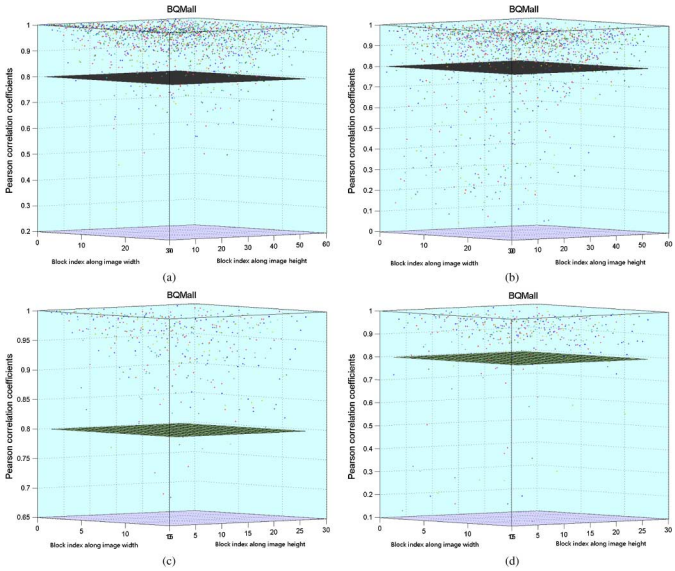

Fig. 7. Plot of the Pearson correlations among subsamples of the 16×16 and 32×32 blocks, respectively. (a) Original 16×16 blocks, $(\mathrm{PCC} \geq 0.8) = 93\%$, (b) Residual 16×16 blocks, $(\mathrm{PCC} \geq 0.8) = 80\%$, (c) Original 32×32 blocks, $(\mathrm{PCC} \geq 0.8) = 97\%$, (d) Residual 32×32 blocks, $(\mathrm{PCC} \geq 0.8) = 89\%$.

TABLE II
CORRELATIONS OF THE INTRA RESIDUAL BLOCKS ($\mathrm{PCC} \geq 0.8$)

Test Sequences	Block Types		
	8×8	16×16	32×32
Akiyo(CIF)	62%	66%	84%
Foreman(CIF)	60%	62%	65%
BlowingBubbles	47%	50%	53%
BasketballDrill	46%	46%	53%
BigShips	70%	77%	85%
City	83%	90%	94%
Sailormen	65%	67%	82%
Average	62%	66%	73%

correlation among the subsamples increases with the transform block sizes. For example, the PCC results of the original "BQMall" are 8×8 block, $(\mathrm{PCC} \geq 0.8) = 87\%$, 16×16 block, $(\mathrm{PCC} \geq 0.8) = 93\%$ and 32×32 block, $(\mathrm{PCC} \geq 0.8) = 97\%$. The PCC results of the "BQMall" residual are 8×8 block, $(\mathrm{PCC} \geq 0.8) = 66\%$, 16×16 block, $(\mathrm{PCC} \geq 0.8) = 80\%$ and 32×32 block, $(\mathrm{PCC} \geq 0.8) = 89\%$. In addition, Table II shows more PCC results of different sequences, where the first ten frames of intra residuals are used in the experiments. One can see that the larger the transform block size, the higher is the correlation among the subsamples, hence the higher efficiency of the proposed transform kernels, which can be demonstrated by the experimental results of block selection ratios in Section IV-C.

B. Spiral-Scanning Table

In this subsection, we study the distributions of transform coefficients produced by (13). The original "BQMall" frame and its residuals are encoded by the proposed CAT algorithm, and experimental results for the AC coefficients of the first macroblock are shown in Fig. 8. As shown in Figs. 8(b) and (d), it can be seen that the proposed method has a better energy compaction performance than that of the DCT. Furthermore, we observed that transform coefficients of the proposed method (13)

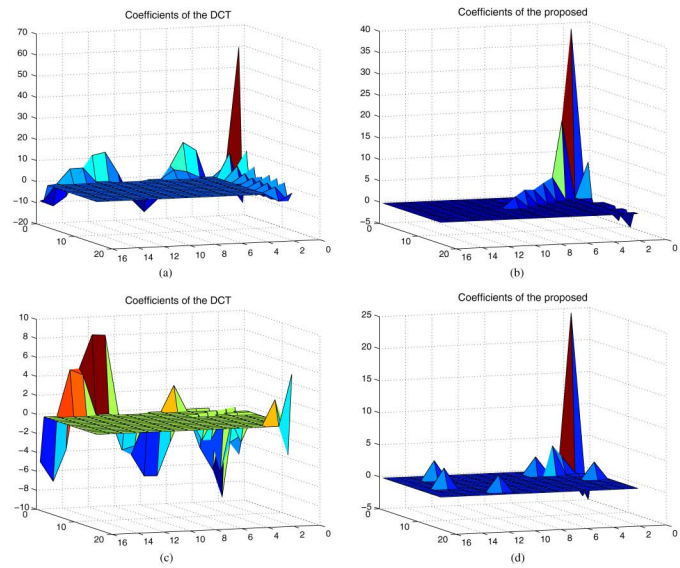

Fig. 8. Distributions of the DCT coefficients and the proposed method in a 16×16 blocks (only AC coefficients). (a) and (b) are obtained from the original "BQMall" frame. (c) and (d) are obtained from its intra residual.

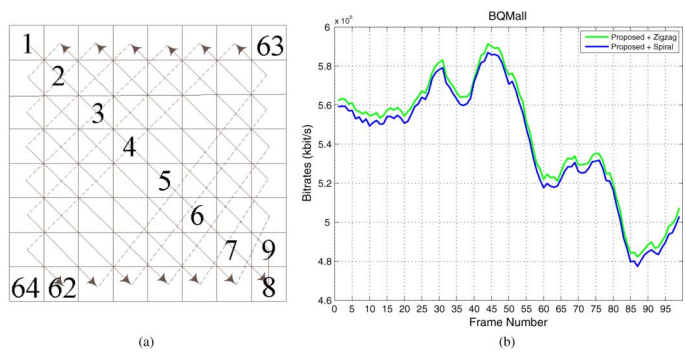

Fig. 9. Illustration of the spiral-scanning table. (a) spiral; (b) performance of the proposed CAT with/without the spiral-scanning method.

are distributed along the diagonal line. Therefore, we propose a new spiral-scanning method for the CAT algorithm to efficiently reorder the quantized coefficients as shown in Fig. 9(a). Fig. 9(b) illustrates the simulation results of "BQMall" under all-intra coding configuration at $\mathrm{QP} = 22$, where a total 99 frames are used in the experiments and both the PSNR results are about 41.6 dB. It can be seen that the proposed spiral-scanning method achieves a lower bitrate than the zigzag scanning method since it is able to extend the zero run-length, thus improving the entropy coding efficiency.

C. Transform Block Size and Block Selection Ratio

To show the effectiveness of the proposed CAT framework, we investigate the selection ratio of a transform block size under all-intra coding configuration in this subsection. The experimental conditions are the same as in Section V, and the selection ratio is computed as:

$$Selectionratio = \frac{\#N \times N\,(Proposed)}{\#N \times N\,(Total)} \times 100\% \qquad (17)$$

TABLE III
COMPARISON RESULTS OF DIFFERENT TRANSFORM BLOCK SIZE UNDER ALL-INTRA CODING CONFIGURATION ($QP = 22, 27, 32, 37$)

Test Sequences	Only TransBlk 8×8		Only TransBlk 16×16	
	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)
Coastguard(CIF)	0.80	-10.87	0.59	-8.28
Container(CIF)	0.78	-10.36	0.36	-5.12
Football(CIF)	0.66	-8.20	0.07	-1.17
Foreman(CIF)	0.35	-6.03	0.26	-3.50
Mobile(CIF)	1.26	-10.9	0.81	-7.16
BasketballPass(W-VGA)	0.29	-4.69	0.28	-4.02
BlowingBubbles(W-VGA)	0.55	-8.28	0.31	-4.67
BQSquare(W-VGA)	0.73	-7.83	0.65	-6.89
Average	0.67	-8.40	0.40	-4.97

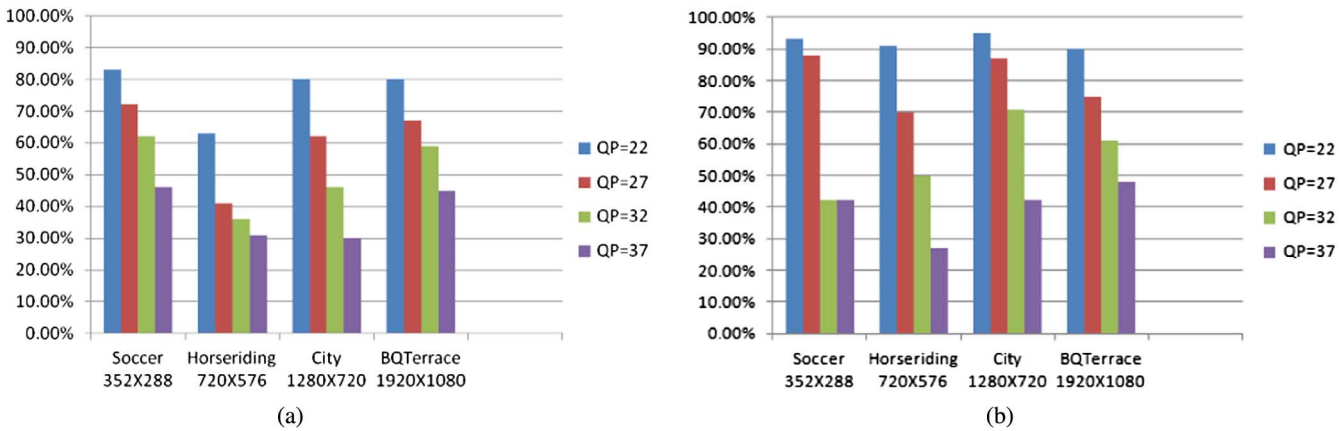

Fig. 10. The selection ratio for a transform block size. (a) 8×8; (b) 16×16.

where $N \times N(Proposed)$ is the number of $N \times N$ blocks coded by proposed method; $N \times N(Total)$ is the total number of $N \times N$ blocks.

As shown in Fig. 3, the proposed CAT framework is implemented in the KTA intra encoder. According to the quad-tree based parsing process, the proposed method involves two transform block sizes, i.e., 8×8 and 16×16. The reason is that the small blocks (i.e., 4×4) may cost too many overhead bits. Furthermore, the correlations among the subsamples of the 4×4 blocks make it not worthwhile to compute the adaptive transform kernels.

Table III shows the performance of the proposed method for two transform block sizes. It can be seen that good coding performance is achieved for all test sequences. For example, an average 0.67 dB Bjontegaard-Delta PSNR (BD-PSNR) for the 8×8 blocks and 0.40 dB BD-PSNR for the 16×16 blocks are achieved, respectively. BD-PSNR is derived from simulation results with various QPs $= \{22, 27, 32, 37\}$. For sequences with rich textures, like "Coastguard" and "Mobile", the proposed method is able to achieve very high coding gain.

In Fig. 10, four sequences with different resolutions are used to evaluate the usage of our method for various block sizes. It should be pointed out that the proposed method is evaluated at the optimal rate distortion (RD) condition. As shown in Fig. 10, high selection ratio is achieved at higher bitrate. The results support our QP scheme (14) in Section III-B, where high quality $X^*_{k,1}$ can be used to derive better transform kernels. In addition, high selection ratio for the big transform blocks can also be observed in Fig. 10(a) and 10(b). It can be seen that the selection ratio of 16×16 is higher than that of 8×8 in most cases (e.g., 13/16). The reason is that the correlation among subsamples of 16×16 is higher than that of 8×8, which indicates better transform kernels can be derived.

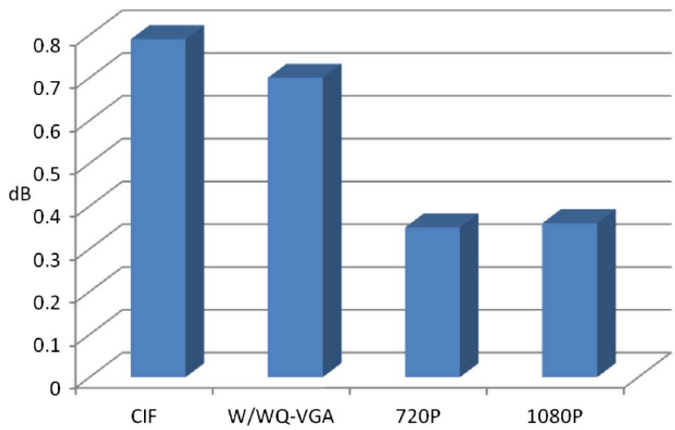

Fig. 11. Coding gain comparison of the DCT and the proposed method.

D. Coding Gain Analysis

The transform coding gain is considered as a benchmark to measure the transform performance. It is usually defined as the ratio of the arithmetic mean of variances of the transform coefficients and the geometric mean of the variances:

$$G_{TC} = \frac{\frac{1}{J}\sum_{j=1}^{J}\sigma_i^2}{\left(\prod_{j=1}^{J}\sigma_j^2\right)^{1/J}} \tag{18}$$

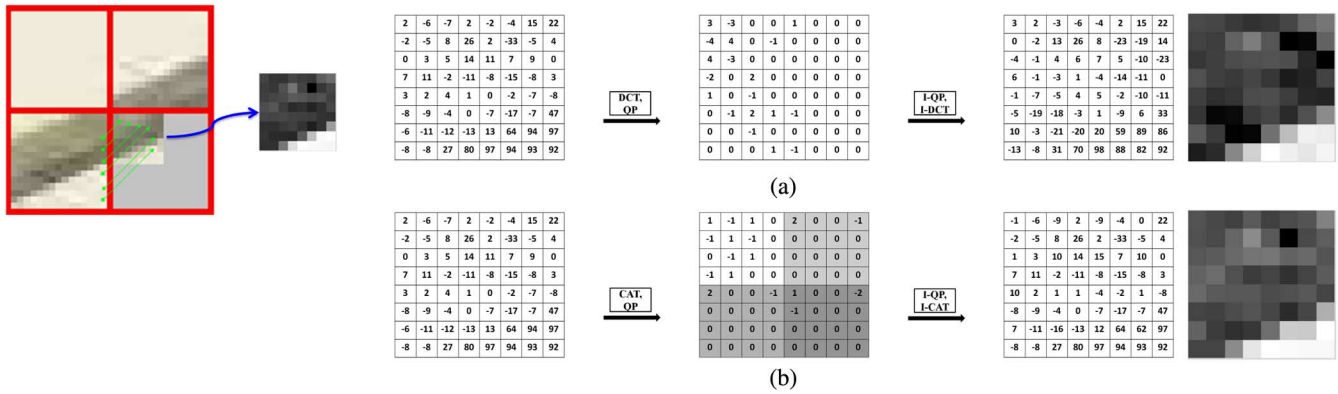

Fig. 12. Comparison of the transform coefficients between the DCT (H.264/AVC) and the proposed method. The most left encoding block is the 29th Macroblock (MB) in the first frame of the "Foreman (CIF)" sequence, and the current predictive residual block are shown in the right-hand side; (a) is the result of the DCT ($\mathrm{MSE} = 62.39$); (b) is the results of the proposed CAT ($\mathrm{MSE} = 26.20$).

where σ_j is the variance of the jth transform coefficient; and J is the total number of the transform coefficients.

In the 2D case, the variances of each transform coefficient are defined as:

$$\begin{aligned}\sigma_{i,j}^2 &= \mathrm{E}\left\{C\left(i,j\right)\cdot C\left(i,j\right)\right\}\\ &= \mathrm{E}\left\{\left[\sum\nolimits_{m,n=0}^{N-1} U_v\left(i,m\right) X_{N\times N}\left(m,n\right) U_h\left(j,n\right)\right]\right.\\ &\quad\left.\times\left[\sum\nolimits_{p,q=0}^{N-1} U_v\left(i,p\right) X_{N\times N}\left(p,q\right) U_h\left(j,q\right)\right]\right\}\\ &= \sum\nolimits_{m,n,p,q=0}^{N-1} \mathrm{E}\left\{X_{N\times N}\left(m,n\right)\cdot X_{N\times N}\left(p,q\right)\right\}\\ &\quad\times U_v\left(i,m\right) U_h\left(j,n\right) U_v\left(i,p\right) U_h\left(j,q\right), \end{aligned} \tag{19}$$

where $\mathrm{E}\{\cdot\}$ is the expectation operator. Furthermore, for discrete-time stationary Markov-1 signals, a separable 2D auto-covariance model is given by:

$$\mathrm{E}\left\{X_{N\times N}\left(m,n\right)\cdot X_{N\times N}\left(p,q\right)\right\} = \rho_1^{|m-p|}\rho_2^{|n-q|}. \tag{20}$$

Using this separable model, the 2D-DCT can be derived as the vertical and horizontal correlations ρ_1 and ρ_2 approaches 1, respectively.

In Fig. 11, we show the average transform coding gain of the proposed method under all-intra configuration. It can be seen that the proposed CAT outperforms the H.264/AVC. In addition, it is observed that the average transform coding gains of the proposed method decreases as the frame resolution changes from 416×240 to 1920×1080.

Fig. 12 shows a further coding gain comparison between the H.264/AVC and the proposed method in an intra coding experiment. The input residual block is from the 29th macroblock of the "Foreman(CIF)" sequence with the intra direction $MODE = 8$. As shown in Fig. 12(a), it is observed that the DCT coefficients are sparsely distributed from the low frequencies to the high frequencies. In comparison with the H.264/AVC, the proposed method is shown in Fig. 12(b), where 4×4 blocks with four different colors denote the quantized transform coefficients of $\{X_{k,1}, X_{k,2}, X_{k,3}, X_{k,4}\}$, respectively. One can see that the proposed method outperforms the H.264/AVC in terms of less coefficients and hence distortion.

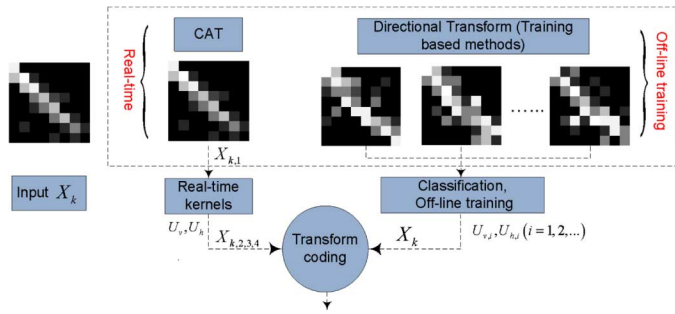

Fig. 13. Comparison of the proposed method and training based methods.

E. Comparison of the Proposed Method and the Directional Transform Methods

Fig. 13 shows the comparison between the proposed method and training-based directional transforms [16], [24], [29]. For the directional transform methods, the transform kernels are off-line trained on a large residual data set. In this situation, directional transforms can be considered as a separable 2D-KLT with a super-large M in (7). Bear in mind that all of these processes are completed off-line and the transform kernels are used for different video sequences. In comparison with the directional transforms, the proposed CAT method obtains the transform kernels adaptively from the reconstructed block. As shown in Fig. 13, the CAT method derives the transform kernels (U_v, U_h) from the block $X_{k,1}^*$. Due to a relatively high correlation between block $X_{k,1}$ and the other three blocks ($X_{k,2}$, $X_{k,3}$ and $X_{k,4}$), the transform kernels (U_v, U_h) have superior energy compaction capability when applied to the other three blocks.

F. Additional Syntax

For each $2N\times 2N$ block, a $CATMode$ flag is used to notify whether the proposed method is used. In addition, code block pattern (CBP) is used to indicate whether non-zero coefficients are sent, and CABAC entropy coding method is used to collect the total bitrate consumption. For example, the overhead bitrate is about 0.2% for "BQMall" and 0.19% for "Container" (see Fig. 14).

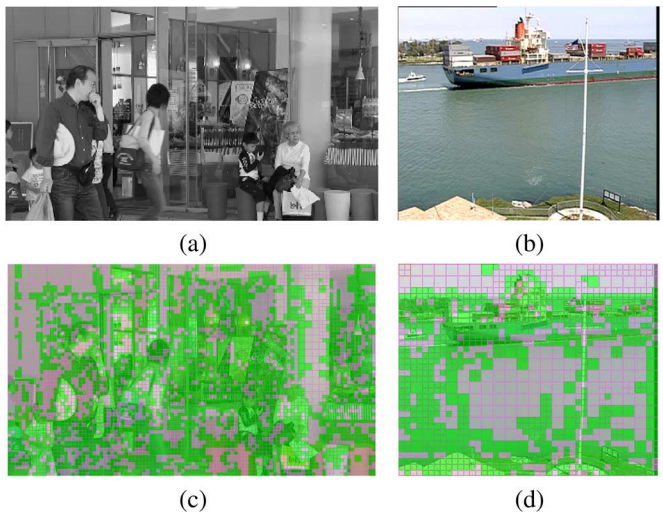

Fig. 14. Illustration of block selections under the all-intra configuration. (a) "BQMall" 832×480; (b) "Container" 352×288; (c) "BQMall" coded by the proposed method; (d) "Container" coded by the proposed method; Green blocks are coded by our method and grey blocks are coded the H.264/AVC.

TABLE IV
TEST CONDITIONS

Test sequence	CIF, W/WQ-VGA, 720p, 1080p
Total frames	99
Benchmark	H.264/AVC High Profile
Coding Structure	All-intra (III...) Low-delay (IPPP...)
Entropy Coding	CABAC
RD Optimization	On
Search range	$\pm$ 64
Reference number	4
QP	I(22, 27, 32, 37) P(23, 28, 33, 38)

V. EXPERIMENTAL RESULTS

A. Test Conditions

In this section, we evaluate the performance of the proposed method against the state-of-the-art methods in the H.264/AVC framework. The proposed method is implemented on the KTA reference software (JM 11.0 KTA2.6r1) [7] and Table IV gives the general test conditions. In the experiments, the transform block sizes are set as 8×8 and 16×16. There are a total of 41 standard sequences of different resolutions, including CIF, W/WQ-VGA, 720p and 1080p. The experiments are conducted on a dual-core (i3-2100@3.10 GHz) workstation with RAM 4 GB that is also used to measure the computational complexity of the proposed method. We measure the performance by the Bjontegaard-Delta Bit Rate (BD-BR) and the Bjontegaard-Delta PSNR (BD-PSNR) metrics [32]. These metrics give the average performance differences between the H.264/AVC and proposed methods. Note that positive BD-PSNR indicates PSNR improvement while negative BD-BR indicates bit rate reduction. The proposed method is compared with several state-of-the-art methods, including Ye's method [16], Lim's [17], Yeo's method [19], Biswas's [23], Zhao's [24] and Dong's [29] in terms of the above metrics.

B. Comparison of Block Selections

Fig. 14 shows two examples of the distributions of blocks encoded by the proposed CAT method. Figs. 14(a) and 14(b) are the original frames of the "BQMall" (832×480) and "Container" (352×288), respectively. Figs. 14(c) and 14(d) show the blocks encoded by the proposed method, where the green blocks are encoded by the CAT method and the grey blocks are encoded by the H.264/AVC at $\mathrm{QP} = 32$. One can see that the blocks with rich textures are frequently encoded using the proposed method.

C. Bjontegaard-Delta Metric Results in the All-Intra Configuration

Table V shows the coding performance of the proposed method and other state-of-the-art methods under all-intra configuration. Bold number indicates the best method and if there is no bold number in a row, it means the H.264/AVC is the best one. From Table V, it can be observed that the proposed method achieves 38 best results from 41 cases. In addition, it can be seen that our method can achieve significant coding gains ranging from 0.76% to 13.61%. This means that to get the same PSNR, our method can save up to 13.61% bitrate compared to the H.264/AVC high profile.

From Table V, comparing to H.264/AVC, Ye's method [16] achieves about 0.22 dB gain of BD-PSNR or 3.72% reduction of BD-BR on average, Yeo's method [19] achieves about 0.15 dB gain of BD-PSNR or 2.79% reduction of BD-BR on average, while our method achieves about 0.55 dB gain of BD-PSNR or 7.95% reduction of BD-BR on average. Based on the above analysis, it can be concluded that the proposed CAT method outperforms the best currently available methods under the all-intra configuration.

D. Bjontegaard-Delta Metric Results in the Low-Delay Configuration

In this subsection, we extended the proposed content adaptive transform framework to the motion compensation (MC) residual coding. In the low-delay configuration, we compare the proposed method with Lim's [17] and Dong's [29] method. In the experimental results, the test condition is set as in [29] for fair comparison.

Table VI shows the comparison results between the H.264/AVC and the best currently available methods under the low-delay configuration. Compared to the H.264/AVC, Dong's method [29] achieves about 0.17 dB gain of BD-PSNR or 5.62% reduction of BD-BR on average, while the proposed method achieves an average 0.34 dB gain of BD-PSNR or 7.0% reduction of BD-BR. On the other hand, Lim's method [17] shows little coding gain.

Finally, the computational complexities of the three methods are summarized as follows. The complexity of Lim's method [17], the proposed method and Dong's method [29] are about 105%, 110% and 123% respectively, as compared to the H.264/AVC. From Table VI, one can see that the proposed CAT method is able to achieve slightly better or comparable performance comparing with state-of-the-art methods under the low-delay configuration.

TABLE V
PERFORMANCE COMPARISON BETWEEN THE PROPOSED METHOD AND STATE-OF-THE-ART METHODS UNDER THE ALL-INTRA CONFIGURATION

Test Sequences	Size	Coding Methods					
		Ye's [16]		Yeo's [19]		Proposed	
		BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)
Akiyo	352×288	0.26	-3.94	0.13	-1.99	**0.56**	**-7.93**
Coastguard	352×288	0.33	-4.86	0.19	-2.77	**0.87**	**-11.24**
Container	352×288	0.16	-2.29	0.05	-0.79	**0.82**	**-11.07**
Football	352×288	0.42	-6.1	-0.02	0.39	**0.52**	**-7.24**
Foreman	352×288	0.22	-3.88	0.1	-1.68	**0.59**	**-10.03**
Mobile	352×288	0.14	-1.34	0.34	-3.05	**1.54**	**-12.85**
Mother-daughter	352×288	0.20	-3.70	0.21	-3.84	**0.19**	**-3.26**
News	352×288	0.37	-4.89	0.18	-2.32	**0.84**	**-10.3**
Salesman	352×288	0.20	-3.24	0.04	-0.84	**0.68**	**-10.18**
Silent	352×288	0.18	-3.21	-0.03	0.59	**0.53**	**-8.91**
Soccer	352×288	0.21	-3.86	0.12	-2.21	**0.61**	**-9.61**
Stefan	352×288	0.42	-4.44	0.10	-1.01	**1.40**	**-13.16**
Tempete	352×288	0.30	-3.59	0.23	-2.55	**1.30**	**-13.61**
waterfall	352×288	0.42	-6.32	-0.07	1.12	**0.62**	**-8.84**
Average		0.27	-0.08	0.11	-1.49	**0.79**	**-9.87**
BasketballPass	416×240	0.19	-3.38	-0.08	1.33	**0.50**	**-7.86**
BlowingBubbles	416×240	0.20	-3.18	0.13	-2.03	**0.79**	**-11.44**
BQSquare	416×240	0.22	-2.54	0.32	-3.47	**1.33**	**-13.44**
Basketball	720×576	0.47	-5.31	0.36	-3.98	**1.32**	**-13.36**
Horseriding	720×576	0.32	-5.31	0.29	-5.32	**0.32**	**-5.13**
Src5-ref	720×576	0.26	-3.49	0.08	-0.83	**0.52**	**-6.39**
BasketballDrill	832×480	0.28	-5.30	0.11	-2.08	**0.44**	**-8.15**
BasketballDrillText	832×480	0.16	-2.84	0.13	-2.27	**0.45**	**-7.69**
BQMall	832×480	0.14	-2.35	0.09	-1.40	**0.61**	**-9.17**
Average		0.20	-3.18	0.16	-2.23	**0.70**	**-9.18**
Bigship	1280×720	0.22	-4.48	0.11	-2.05	**0.39**	**-7.34**
City	1280×720	0.30	-4.53	0.18	-2.81	**0.48**	**-6.34**
Crew	1280×720	0.11	-2.75	0.19	-5.26	**0.12**	**-3.12**
Cyclists	1280×720	0.24	-5.93	**0.36**	**-8.99**	0.12	-2.94
Harbour	1280×720	0.42	-5.69	**0.50**	**-7.11**	0.40	-4.91
Jet	1280×720	0.14	-3.86	-0.08	1.23	**0.15**	**-4.24**
Night	1280×720	0.30	-4.29	0.45	-13.21	**0.54**	**-7.24**
Optis	1280×720	0.32	-6.36	0.17	-3.80	**0.28**	**-5.18**
Panslow	1280×720	0.14	-2.65	0.02	-0.06	**0.48**	**-10.23**
Raven	1280×720	0.27	-5.12	0.18	-3.57	**0.19**	**-3.50**
Sailormen	1280×720	0.16	-2.99	0.12	-2.27	**0.39**	**-6.92**
Sheriff	1280×720	**0.38**	**-6.34**	0.17	-3.04	0.37	-5.76
Spincalendar	1280×720	0.25	-4.15	0.16	-2.74	**0.68**	**-10.63**
Average		0.25	-4.55	0.19	-4.13	**0.35**	**-6.03**
BasketballDrive	1080×1920	-0.06	1.79	0.09	-1.51	**0.13**	**-4.08**
BQTerrace	1920×1080	0.25	-3.92	0.31	-10.36	**0.85**	**-11.94**
Cactus	1920×1080	0.21	-5.15	-0.03	0.59	**0.39**	**-8.86**
Kimono1	1920×1080	0.11	-3.17	0.32	-9.52	**0.02**	**-0.76**
ParkScene	1920×1080	0.26	-5.4	-0.05	1.03	**0.40**	**-8.00**
Average		0.15	-3.17	0.05	-1.52	**0.36**	**-6.73**
Overall		0.22	-3.72	0.15	-2.79	**0.55**	**-7.95**

TABLE VI
PERFORMANCE COMPARISON BETWEEN THE PROPOSED METHOD AND STATE-OF-THE-ART METHODS UNDER THE LOW-DELAY CONFIGURATION

Sequence	Coding Methods					
	Lim's [17]		Dong's [29]		Proposed	
	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)	BD-PSNR (dB)	BD-BR (%)
Bigship	0.00	0.08	0.15	-5.00	0.45	-10.15
City	-0.02	0.48	0.18	-5.64	0.50	-9.91
Crew	0.00	0.21	0.38	-13.19	0.26	-6.34
Harbour	0.01	-0.07	0.11	-2.83	0.29	-6.58
Jet	0.01	-0.13	0.09	-4.15	0.37	-4.26
Night	0.01	0.12	0.13	-3.26	0.25	-5.27
Raven	-0.01	0.17	0.31	-7.72	0.29	-4.21
Sailormen	0.00	0.16	0.13	-4.78	0.37	-9.87
Sheriff	-0.01	0.21	0.15	-5.09	0.28	-7.08
ShutterStart	-0.02	0.62	0.15	-5.54	0.35	-3.77
Spincalendar	-0.01	0.23	0.13	-4.66	0.43	-9.57
Average	**0.00**	**0.12**	**0.17**	**-5.62**	**0.34**	**-7.0**
Complexity	105 %		123 %		110 %	

E. Rate Distortion Plot and Visual Quality

Fig. 15 and Fig. 16 show the rate distortion (RD) curves under the all-intra and low-delay configurations, respectively. The x-axis is the average bitrate of each method at $QP = \{22, 27, 32, 37\}$. Correspondingly, y-axis is the average PSNR at each QP. Since RD curves directly show the coding performance over the entire QP range, several best available methods are evaluated, such as Ye's method [16], Lim's [17], Yeo's [19], Biswas's [23], Zhao's [24] and Dong's [29]. From Fig. 15 and Fig. 16, it can be observed that the H.264/AVC gives the lowest coding gain whereas the CAT method achieves better or comparable results as compared to the others. By taking the RD curves of the H.264/AVC as the benchmark, it can be seen that the proposed method can significantly improve the coding efficiency under the all-intra and low-delay configurations.

Fig. 17 shows the visual quality results of the "Foreman (CIF)" sequence under the all-intra coding configuration on

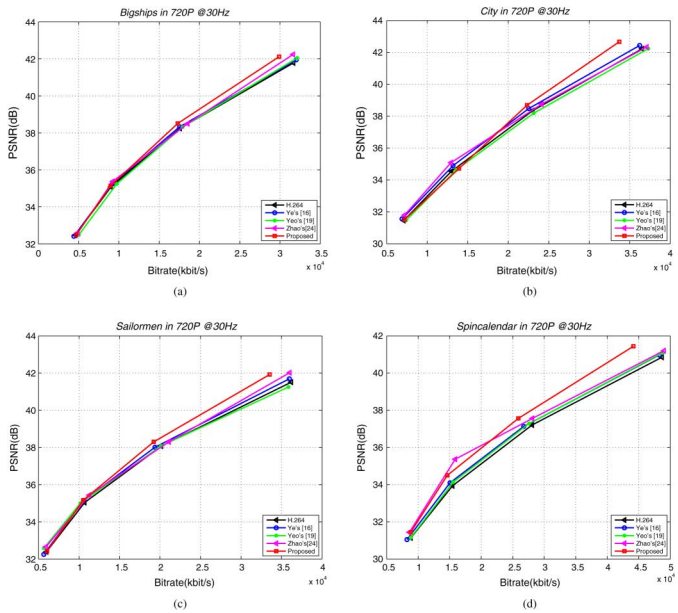

Fig. 15. Rate distortion curves under the all-intra configuration.

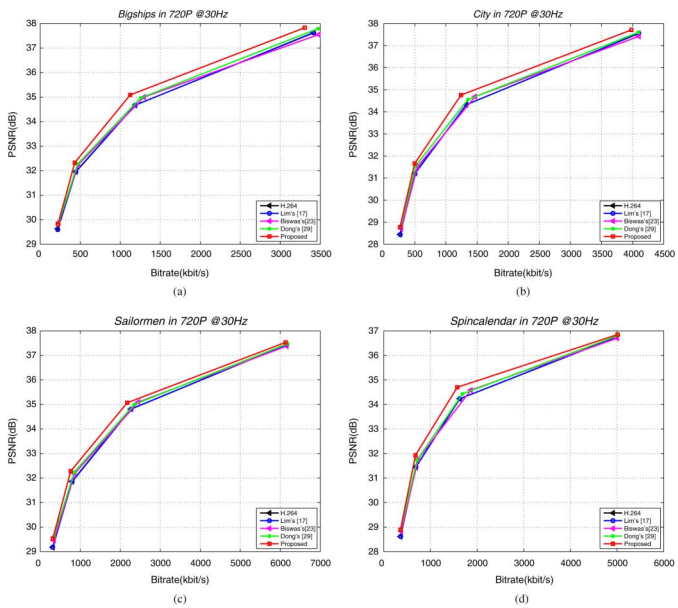

Fig. 16. Rate distortion curves under the low-delay configuration.

the KTA platform. The first column shows the result of the H.264/AVC and its cropped region of about 5×5 zoom for reference, respectively. Correspondingly, the second and third columns show the results of Ye's method [16] and Yeo's [19], respectively. For comparison, the subjective performance of our method is shown in the fourth column. As shown in Fig. 17, there are visible differences in some regions of the decoded frame although the improvements are not obvious. For example, in Fig. 17(d), the texture in the red rectangle is visually more pleasing than those of Figs. 17(a), 17(b) and 17(c). Zoom-in parts are shown in Figs. 17(e), 17(f), 17(g) and 17(h).

TABLE VII
COMPUTATIONAL COMPLEXITY OF THE ENCODER

Sequences 99 frames	QP	Total Coding Time(s)			
		Anchor	Ye's [16]	Yeo's [19]	Proposed
CIF Average Time (in seconds)	22	55	85	74	122
	27	45	67	61	106
	32	37	56	44	88
	37	32	48	37	70
W/WQ-VGA Average Time (in seconds)	22	169	253	215	382
	27	137	206	175	311
	32	113	172	149	248
	37	98	146	125	207
720p average time (in seconds)	22	440	672	583	1045
	27	359	544	475	857
	32	304	469	404	675
	37	273	422	357	528
1080p average time (in seconds)	22	1019	1499	1471	2400
	27	814	1210	1188	1927
	32	685	1034	1004	1498
	37	621	923	919	1159
Average complexity		100%	150%	130%	220%

TABLE VIII
COMPUTATIONAL COMPLEXITY OF THE DECODER

Decoding Time Complexity				
QP	22	27	32	37
Complexity	105%	106%	109%	112%
Average	108%			

F. Computational Complexity

In H.264/AVC, the complexity is mainly from rate distortion optimization (RDO) process that is used to choose the best coding modes. Since the proposed method is implemented in the quad-tree structures of the KTA encoder, the additional complexity is increased in the encoder. The additional complexity is mainly from deriving the separable 2D-KLT transform kernels, and we define that one KLT operation denotes the operations of addition, subtraction, multiplication or division for calculating transform kernels [30] once. When the RDO tool is on, the additional complexity is relatively high in the encoder. However, it is worth noting that we only need to calculate transform kernels for an N $\times$ N block when an input block is 2N $\times$ 2N, and the decoder complexity will only be one operation because mode selection is only carried out in the encoder.

The complexity of the proposed method consists of two parts: (1) calculating the transform kernels and (2) calculating the transform coefficients of the separable 2D-KLT (13). In general, when the input block is $M \times N$ ($N \geq M$), the operations are: (1) calculating the transform kernels $=4M^2N + 8MN^2 + 9N^3$, and (2) calculating the transform coefficients $= 4M^2N - M^2 - NM$. Thus, the total operation for the proposed method is $8(M^2N + MN^2) + 9N^3 - M^2 - NM$.

Table VII summarizes the average encoding time. It can be seen that the encoding complexity of the proposed method is about twice that of the H.264/AVC. It is also observed that the proposed method is more complex than Ye's [16] and Yeo's [19]. It is clear that the additional complexity is from the operations of updating the adaptive transform kernels. In addition, Table VIII shows the complexity of the proposed method in the decoder side. The average decoding complexity of the proposed method compared to the KTA decoder is 108% under the all-intra configuration. One can see that the proposed method can be efficiently used in video coding because the increased execution time is acceptable (e.g., 120% in the encoder and 8% in the decoder).

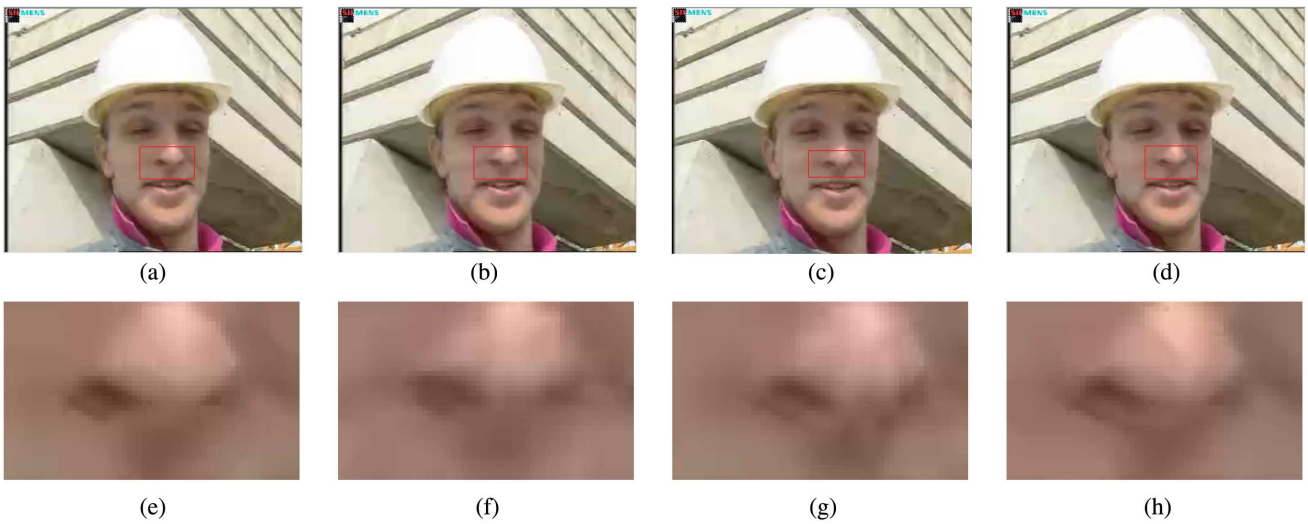

Fig. 17. Subjective coding performance comparison of the first frame in "Foreman (CIF)". (a) is the result of the H.264/AVC and the red rectangle part is amplified in (e); (b) is the result of Ye's [16] and the red rectangle part is amplified in (f); (c) is the result of Yeo's [19] and the red rectangle part is amplified in (g); (d) is the result of the proposed CAT and the red rectangle part is amplified in (h).

VI. Conclusions

In this paper, we introduce a content adaptive transform method for the H.264/AVC-based video coding. Based on pixel rearrangement, we construct a novel transform coding framework that utilizes adaptive transform kernels to decorrelate the prediction residuals. Unlike the traditional adaptive transforms, the proposed method does not need to transmit the transform kernels to the decoder, which greatly saves the overhead bitrate. In addition, to improve the coding performance, we propose a spiral-scanning method to reorder the quantized coefficients before the RLC and entropy coding. Experimental results show that the proposed method achieves an average bitrate reduction of 7.95% and 7.0% under all-intra and low-delay configurations, respectively, as compared to the H.264/AVC high profile.

We believe that the proposed method can be further extended in a number of directions. For example, the proposed encoding framework can be extended to the large transform block sizes, such as 32×32 and 64×64. In addition, the proposed method can be used in the rectangular blocks, which will benefit the rectangular quad-tree structure, and early mode decision algorithms will be designed to avoid the redundant computation of adaptive transform kernels. Also, the proposed method can be directly used in image compression, where high correlation is able to improve its performance.

References

[1] T. Smith and J. Guild, "The CIE colorimetric standards and their use," *Trans. Opt. Soc.*, vol. 33, no. 3, p. 73, 2002.

[2] C. Kim and C. C. J. Kuo, "Feature-based intra-/intercoding mode selection for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 441–453, 2007.

[3] A. C. Tsai, A. Paul, J. C. Wang, and J. F. Wang, "Intensity gradient technique for efficient intra-prediction in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 694–698, 2008.

[4] Y. Piao and H. W. Park, "Adaptive interpolation-based divide-and-predict intra coding for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1915–1921, 2010.

[5] A. K. Jain, "A sinusoidal family of unitary transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 4, pp. 356–365, 1979.

[6] H. S. Malvar, M. Hallapuro, A. 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, 2003.

[7] K. Sühring, G. Heising, and D. Marpe *et al., KTA(2.6r1) Reference Software*, 2009 [Online]. Available: http://iphome.hhi.de/suehring/tml/down load

[8] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[9] 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, 2003.

[10] L. Yu, S. J. Chen, and J. P. Wang, "Overview of AVS-video coding standards," *Signal Process.: Image Commun.*, vol. 24, no. 4, pp. 247–262, 2009.

[11] K. R. Rao and P. Yip, *The Transform and Data Compression Handbook*. Boca Raton, FL, USA: CRC, 2000.

[12] M. H. Wang and K. N. Ngan, "An efficient content adaptive transform for video coding," in *Proc. IEEE China Summit and Int. Conf. Signal and Information Processing (ChinaSIP)*, 2013, pp. 547–550.

[13] M. H. Wang, K. N. Ngan, and H. Q. Zeng, "A rate distortion optimized transform for motion compensation residual," in *Proc. Picture Coding Symp. (PCS2013)*, 2013, pp. 13–16.

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

[15] C. L. Chang, M. Makar, S. S. Tsai, and B. Girod, "Direction-adaptive partitioned block transform for color image coding," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1740–1755, 2010.

[16] 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, (ICIP)*, 2008, pp. 2116–2119.

[17] S. C. Lim, D. Y. Kim, and Y. L. Lee, "Alternative transform based on the correlation of the residual signal," in *Proc. IEEE Congr. Image and Signal Processing, 2008 CISP'08*, 2008, vol. 1, pp. 389–394.

[18] J. N. Han, A. Saxena, V. Melkote, and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1874–1884, 2012.

[19] C. Yeo, Y. H. Tan, Z. Li, and S. Rahardja, "Mode-dependent transforms for coding directional intra prediction residuals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 545–554, 2012.

[20] A. Dapena and S. Ahalt, "A hybrid DCT-SVD image-coding algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 2, pp. 114–121, 2002.

[21] Z. Gu, W. Lin, B. Lee, and C. T. Lau, "Low-complexity video coding based on two-dimensional singular value decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 674–687, 2012.

[22] C. Ding and J. Ye, "2-dimensional singular value decomposition for 2D maps, and images," in *Proc. SIAM Int. Conf. Data Mining*, 2005, pp. 32–43.
[23] M. Biswas, M. R. Pickering, and M. R. Frater, "Improved H.264-based video coding using an adaptive transform," in *Proc. IEEE Int. Conf. Image Processing (ICIP), 2010*, 2010, pp. 165–168.
[24] X. Zhao, L. Zhang, S. Ma, and W. Gao, "Video coding with rate-distortion optimized transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 1, pp. 138–151, 2012.
[25] R. A. Cohen, S. Klomp, A. Vetro, and H. F. Sun, "Direction-adaptive transforms for coding prediction residuals," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2010, pp. 185–188.
[26] O. G. Sezer, R. Cohen, and A. Vetro, "Robust learning of 2-d separable transforms for next-generation video coding," in *Proc. IEEE Data Compression Conf. (DCC), 2011*, 2011, pp. 63–72.
[27] A. Saxena and F. C. Fernandes, "Mode dependent DCT/DST for intra prediction in block-based image/video coding," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2011, pp. 1685–1688.
[28] E. Alshina, A. Alshin, and F. C. Fernandes, "Rotational transform for image, and video compression," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2011, pp. 3689–3692.
[29] J. Dong and K. N. Ngan, "Two-layer directional transform for high performance video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 619–625, 2012.
[30] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, no. 5, pp. 403–420, 1970.
[31] M. H. Wang and B. Yan, "Lagrangian multiplier based joint three-layer rate control for H.264/AVC," *IEEE Signal Process. Lett.*, vol. 16, no. 8, pp. 679–682, 2009.
[32] G. Bjontegard, "Calculation of average PSNR differences between RD-curves," *ITU-T VCEG-M33*, 2001.

Miaohui Wang (S'13) is currently pursuing the Ph.D. degree in the Department of Electronic Engineering, the Chinese University of Hong Kong (CUHK). Before that he received his B.S. degree in Applied Mathematics from Xi'an University of Posts and Telecommunications (XUPT) and M.A. degree from School of Computer Science, Fudan University in 2007 and 2010 respectively. His current research interests cover a wide range of topics related with video coding, computer vision and code optimization, including rate control, transform coding, error concealment and image inpainting.

Mr. Wang received the Best Thesis Award in Shanghai city (2011) and Fudan University (2012), respectively.

King Ngi Ngan (M'79–SM'91–F'00) received the Ph.D. degree in electrical engineering from Loughborough University, Loughborough, U.K. He is currently a Chair Professor with the Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, Hong Kong. He was previously a Full Professor with Nanyang Technological University, Singapore, and with the University of Western Australia, Perth, Australia. He holds honorary and visiting professorships with numerous universities in China, Australia, and South East Asia. He has published extensively, including three authored books, six edited volumes, over 300 refereed technical papers, and has edited nine special issues in journals. He holds ten patents in image or video coding and communications.

Dr. Ngan has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the Journal on Visual Communications and Image Representation, the EURASIP Journal of Signal Processing: Image Communication, and the Journal of Applied Signal Processing. He has chaired a number of prestigious international conferences on video signal processing and communications, and has served on the advisory and technical committees of numerous professional organizations. He co-chaired the IEEE International Conference on Image Processing, Hong Kong, in September 2010. He is a fellow of IET, U.K., and IEAust, Australia, and was an IEEE Distinguished Lecturer from 2006 to 2007.

Long Xu received his M.S. degree in Applied Mathematics from Xidian University, China, in 2002, and his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2009. He was a Postdoc researcher at the Department of Computer Science of City University of Hong Kong, and the Department of Electronic Engineering of Chinese University of Hong Kong from August 2009 to December 2012. Now, he is with both the Rapid-Rich Object Search Lab and School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include image/video coding, wavelet-based image/video coding and computer vision.