Video Compression Artifact Reduction via Spatio-Temporal Multi-Hypothesis Prediction

Xinfeng Zhang, Ruiqin Xiong, *Member, IEEE*, Weisi Lin, *Senior Member, IEEE*, Siwei Ma, *Member, IEEE*, Jiaying Liu, *Member, IEEE*, and Wen Gao, *Fellow, IEEE*

***Abstract*— Annoying compression artifacts exist in most of lossy coded videos at low bit rates, which are caused by coarse quantization of transform coefficients or motion compensation from distorted frames. In this paper, we propose a compression artifact reduction approach that utilizes both the spatial and the temporal correlation to form multi-hypothesis predictions from spatio-temporal similar blocks. For each transform block, three predictions with their reliabilities are estimated, respectively. The first prediction is constructed by inversely quantizing transform coefficients directly, and its reliability is determined by the variance of quantization noise. The second prediction is derived by representing each transform block with a temporal auto-regressive (TAR) model along its motion trajectory, and its corresponding reliability is estimated from local prediction errors of the TAR model. The last prediction infers the original coefficients from similar blocks in non-local regions, and its reliability is estimated based on the distribution of coefficients in these similar blocks. Finally, all the predictions are adaptively fused according to their reliabilities to restore high-quality videos. The experimental results show that the proposed method can efficiently reduce most of the compression artifacts and improve both subjective and objective quality of block transform coded videos.**

***Index Terms*— Compression artifacts, block transform coding, auto-regressive, non-local estimation, multiple hypotheses.**

I. Introduction

ALTHOUGH the state-of-the-art video coding standards, e.g., H.264/AVC and HEVC, have improved the efficiency of image and video compression significantly, they still produce annoying compression artifacts at low bit rates, e.g., blocking artifacts and ringing artifacts. These artifacts not only deteriorate the perceptual quality of video frames remarkably, but also affect the performance of some video applications, such as object detection and recognition [1]. In real-world video service systems, such as YouTube, the video contents are usually compressed at low bit rates due to limited bandwidth and will suffer from annoying artifacts when the videos contain rich textures or complex motions. This leads to poor user experience. Compression artifacts are mainly caused by two sources. Firstly, the coarse quantization applied to each block independently will turn some transform coefficients into zeros and causes discontinuity across block boundaries. Secondly, the motion compensation in inter-frame coding, which copies blocks from a previously reconstructed frames, is likely to propagate the coding artifacts from one frame to another. The block discontinuity within a frame may be even aggravated when two adjacent blocks are predicted from different locations or different reference frames.

Many compression artifact reduction methods have been proposed in the literatures. These methods can be classified into two categories, i.e., in-loop methods and post-processing methods. The in-loop filtering methods process a compressed frame within the coding loop, and the resulting frame can be utilized as reference to provide prediction for the coding of future frames. Examples of this category include the deblocking filter in H.264/AVC [2], Sample Adaptive Offset (SAO) [3] in HEVC, and the widely discussed adaptive loop filter (ALF) [4], [5] in HEVC development. These methods mainly take advantage of the local smoothness in image and adjust the filtering strength according to coding information, such as coding modes, quantization parameters (QP) and rate-distortion cost [6], to achieve a good reconstruction quality. These in-loop filtering methods require a standard compliant decoder to do the same filtering operation in order to synchronize with the encoder.

Besides in-loop filters, post-processing methods also produce promising results by reducing the compression artifacts of a decoded frame outside of coding loop, without specific requirement for being compatible with different video coding standards. The post-processing methods can be easily plugged into a video application system, right after the decoders, to improve the quality of reconstructed videos. This actually helps to reduce the required bandwidth for video transmission. Many post-processing methods are also proposed both in spatial domain and transform domain. Reeve and Lim [7] applied a 3×3 Gaussian filter to the pixels around block boundaries to smooth out the blocking artifacts. Ramamurthi and Gersho [8] utilized nonlinear space-variant filters based on edge-oriented

Manuscript received April 16, 2015; revised August 15, 2015 and September 22, 2015; accepted September 22, 2015. Date of publication October 1, 2015; date of current version November 11, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61370114 and Grant 61322106, the National Basic Research Program of China (973 Program) under Grant 2015CB351800, and the Beijing Natural Science Foundation under Grant 4132039. The work conducted in the Rapid-Rich Object Search (ROSE) Lab was supported by the National Research Foundation, Singapore, under its Interactive Digital Media (IDM) Strategic Research Programme. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jing-Ming Guo. *(Corresponding author: Ruiqin Xiong.)*

X. Zhang and W. Lin are with the Rapid-Rich Object Search Laboratory, Nanyang Technological University, Singapore 639798 (e-mail: xfzhang@ntu.edu.sg; wsling@ntu.edu.sg).

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

J. Liu is with the Institute of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: liujiaying@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/TIP.2015.2485780

classifiers to smooth out blocking artifacts. Buades *et al.* [9] proposed the non-local means (NLM) filter to predict each pixel by a weighted average of its surrounding pixels, where the weights are determined by the similarity of the corresponding image patches located at the source and target coordinates. Takeda *et al.* [10] proposed a signal-dependent steering kernel regression (SKR) framework for denoising. However, the above methods only considered the smoothness or the regularity in pixel intensities, which may smooth out the true edges or texture details, and in some worse cases, further deteriorating the quality of decoded videos. To avoid these problems, Zhai *et al.* [11], [12] utilized the quantization table in compressed stream and standard deviation of image blocks to decide the filtering strength, and employ the quantization intervals to constrain the filtered coefficients.

Besides the image prior of local smoothness in spatial domain, the image prior of sparsity in transform domain is also widely utilized in image processing and related fields [13]–[24]. Wu *et al.* [13] proposed a deblocking algorithm by transforming the decoded image into wavelet domain and adaptively shrinking the wavelet coefficients based on the difference of coefficients in neighboring blocks. Zhai *et al.* [14] proposed an AC regularization method in DCT domain to suppress the blocking artifacts in monotone areas. Foi *et al.* [15] utilized a point wise shape adaptive discrete cosine transform (SA-DCT) to represent image with sparse coefficients and reduce noise by thresholding coefficients. In order to adapt to different image structures, Elad and Aharon [16] proposed to learn an over-complete dictionary via KSVD to get sparser image representation. The famous denoising method, BM3D [17], explores the image self-similarity to cluster non-local similar patches and performing collaborative filtering in a 3D transform domain. Zhang *et al.* [18], [19] estimate original coefficients by combining non-local similar blocks and decoded coefficients adaptively. Dong et al. improves the denoising performance by utilizing PCA to clustered similar patches in CSR [20] and LPG-PCA [21]. When there are not enough similar patches, the performance of these methods may deteriorate since the assumed image prior becomes no longer valid. Therefore, one image prior model can be more efficient than others only under certain conditions, and it may become unsuitable when the conditions are no longer satisfied.

In this paper, we propose a new approach to reduce video compression artifacts by adaptively fusing different coefficient predictions based on the multiple hypotheses in transform domain. This is achieved by estimating original DCT coefficients in all the transform blocks located at any pixel position. For each transform block, three predictions are generated and adaptively fused based on their reliabilities. The first prediction is the coefficients directly decoded from compressed video stream by inverse quantization, and its reliability is determined by quantization steps. The second prediction is acquired by representing each transform block with temporal neighboring blocks along its motion trajectory based on a temporal auto-regressive (TAR) model. Its reliability is estimated from the distribution of local prediction errors of the TAR model. The last predictor infers the original coefficients from non-local similar blocks in the current frame and the neighboring ones. For each block, K-nearest neighbors are collected based on L2 norm of the difference of coefficients in transform blocks, and they are weighted and averaged to generate the prediction according to their similarity with the target block. The reliability of the last prediction is estimated according to the distribution of non-local transform-block coefficients and the quantization noise. In addition, in order to avoid smoothing out textures excessively, we take quantization steps to constrain the estimated coefficients to the same quantization interval as the original coefficients. Since the approach is built on multi-hypothesis estimation based on multiple image prior models, it can adapt to different video content better than the existing methods and can produce reconstructed video with better quality.

The remainder of this paper is organized as follows. In Section II, we first review the basic scheme of video coding and the characteristic of compression noise. Section III formulates the proposed compression artifact reduction framework with multiple hypotheses, and then describes every hypothesis and its reliability in detail. Section IV gives the numerical solution of the proposed method. Experimental results are reported in Section V and Section VI concludes the paper.

II. Analysis of Video Compression Noise

In this section, we firstly briefly review a few concepts and notations relevant to block-transform video coding for the convenience of later discussion of this paper. Then some distribution characteristics of compression noise are analyzed.

A. Basic Concepts and Notations

Suppose we have a video sequence $\{\boldsymbol{x}_t\}$ and $\boldsymbol{x}_t$ is the t^{th} frame with size of $H \times W$, where $\boldsymbol{x}_t(i, j)$ denotes a pixel with vertical and horizontal coordinates i and j, respectively. In block-transform video coding, the input video frame, $\boldsymbol{x}_t$, is divided into a group of non-overlapped blocks (i.e., coding unit, CU) with different sizes. These blocks, denoted as $\boldsymbol{x}_{\mathcal{B}}$, are coded in raster scanning order. The data in each block is predicted from the previous coded frames via motion estimation or the neighboring coded pixels in current frame (e.g., left and upper pixels) along different directions. The prediction residuals in each block are transformed, quantized and entropy coded into the compressed bitstream [25], [26]. We use $\boldsymbol{x}_{\mathcal{B}}$ and $\mathbf{X}_{\mathcal{B}}$ to represent the data (i.e., pixel intensity) and the transform coefficients of a block $\mathcal{B}$, respectively. They are related by the block-DCT (or Discrete Sine Transform, DST) $\mathcal{T}$, $\mathbf{X}_{\mathcal{B}} = \mathcal{T}(\boldsymbol{x}_{\mathcal{B}})$, (inverse transform $\boldsymbol{x}_{\mathcal{B}} = \mathcal{T}^{-1}(\mathbf{X}_{\mathcal{B}})$). The decoded video frames are reconstructed by inverse transformation and quantization. The reconstructed coefficients are,

$$\mathbf{Y}_{\mathcal{B}}(u, v) = \left[\frac{\mathbf{X}_{\mathcal{B}}(u, v) - \mathbf{X}^p_{\mathcal{B}}(u, v)}{Q(u, v)}\right] Q(u, v) + \mathbf{X}^p_{\mathcal{B}}(u, v), \tag{1}$$

where $[x]$ is the operation that rounds the variable x to the nearest integer and $Q(u, v)$ is the quantization step and $\mathbf{X}^p_{\mathcal{B}}(u, v)$ is the prediction of block $\mathcal{B}$ in transform domain. For some special prediction modes, e.g., skip mode and direct

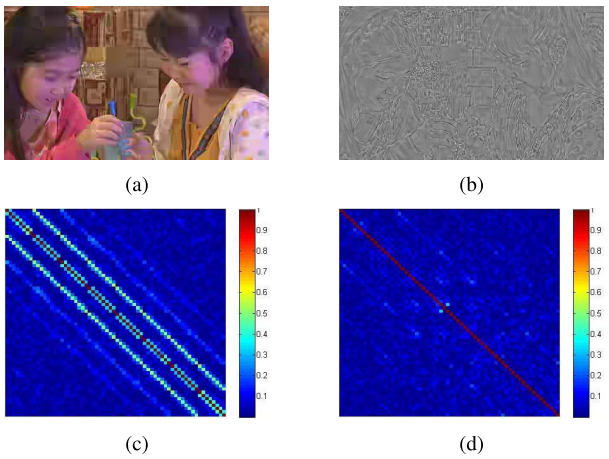


Fig. 1. Characteristic of compression noise. (a) Compressed video frame by HEVC, *BlowingBubbles*, (b) compression noise, (c) the correlation coefficients of compression noise in 8×8 blocks of spatial domain, (d) the correlation coefficients of compression noise in DCT domain.

mode [26], the reconstructed blocks directly copy the previous reconstructed blocks, which not only retain the artifacts in copied blocks but also causes the block discontinuities when two neighboring blocks are predicted from blocks in different frames or nonadjacent positions.

B. Video Compression Noise

Video compression noise is quite different from the some common noise, e.g., Gaussian noise arising during image acquisition or salt-and-pepper noise arising during image transmission, which can be regarded as independent in spatial domain. However, compression noise is correlated in spatial domain due to the similar operations applied to each block with approximate quantization steps. Fig.1(a) illustrates an example of compression noise in sequence *BlowingBubbles* compressed by HEVC intra coding with QP=42. Fig.1(b) directly shows compression noises in spatial domain. From the two figures, we can see that the compression noises are obvious around block boundaries and image edges. In Fig.1(c), we illustrate the absolute values of correlation coefficients of compression noise in 8×8 blocks, which also shows that compression noises are highly correlated in spatial domain as observed in Fig.1(a) and Fig.1(b). Therefore, compression noise is difficult to be reduced while preserving image texture well in spatial domain with traditional low-pass filtering methods, which usually assume noise being uncorrelated or even independent.

Considering the decorrelation property of DCT, we transform compression noise from spatial domain into frequency domain with 8×8 DCT and calculate the correlation coefficients of transformed compression noise. Fig.1(d) shows the absolute values of correlation coefficients among different bands in DCT blocks. We can see that the correlation coefficients among different bands are very small, which shows that compression noise is almost uncorrelated in transform domain. Based on the good statistical characteristic of compression noise in transform domain, we design the compression noise reduction method in DCT domain and estimated original coefficients independently for each band.

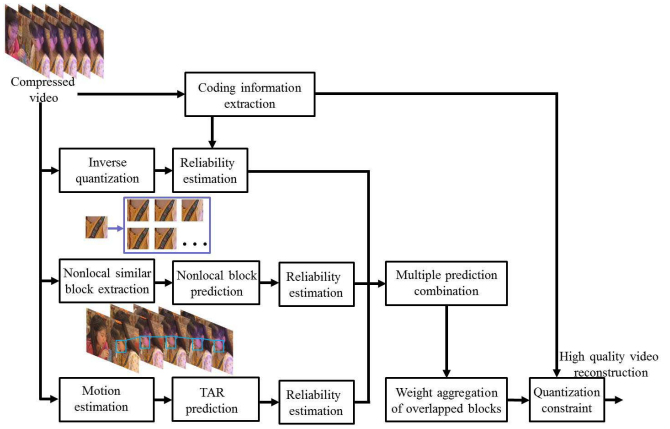


Fig. 2. Framework of the proposed compression artifact reduction method.

III. The Proposed Framework

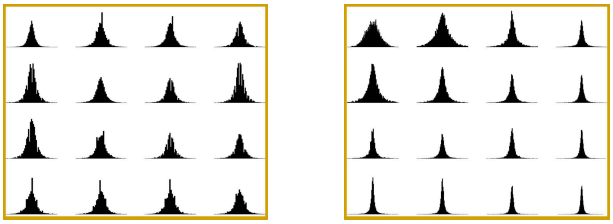
A. Spatial-Temporal Multi-Hypothesis Prediction

In a standard decoder, coded image is reconstructed simply by inversely transforming the quantized coefficients for each coding block. To tackle the compression noises generated from coarse quantization and motion compensation, in this paper, we not only take the reconstructed coefficients from the decoded image as an estimation of the original coefficients, but also utilize the non-local similar transform blocks and the temporal local smoothness to generate another two predictions to restore high quality videos. The final estimated coefficients are determined jointly by the three predictions according to their reliabilities. Here, the reliability of prediction reflects its statistical accuracy for the original coefficients. In order to further reduce compression noises, overlapping blocks are weighted and aggregated to generate high quality videos. The framework of the proposed compression noise reduction method is illustrated in Fig.2.

By introducing three distance metrics, $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, the proposed method is formulated as the following optimization problem,

$$\begin{aligned}\hat{\mathbf{X}}_t = \underset{\mathbf{X}_t}{\operatorname{argmin}} & \sum_{\mathcal{B}_t \in \mathbf{\Omega}} \mathcal{D}_1\left(\mathbf{X}_{\mathcal{B}_t}, \mathbf{Y}_{\mathcal{B}_t}\right) \\ & + \sum_{\mathcal{B}_t \in \mathbf{\Omega}} \mathcal{D}_2\left(\mathbf{X}_{\mathcal{B}_t}, \{\mathbf{Y}_{\mathcal{B}_k}\}_{B_k \in \mathcal{M}(\mathcal{B}_t)}\right) \\ & + \sum_{\mathcal{B}_t \in \mathbf{\Omega}} \mathcal{D}_3\left(\mathbf{X}_{\mathcal{B}_t}, \{Y_{\mathcal{B}_k}\}_{\mathcal{B}_k \in \mathcal{N}(\mathcal{B}_t)}\right). \end{aligned} \quad (2)$$

Here, $\mathcal{M}(\mathcal{B}_t)$ is a block set composed of the blocks along motion trajectory of $\mathcal{B}_t$, and $\mathcal{N}(\mathcal{B}_t)$ is another block set composed of non-local similar blocks in current frame and neighboring frames. $\mathbf{\Omega}$ is the block set composed of all the blocks in the t^{th} frame. The first term, $\mathcal{D}_1$, directly measures the distance between original coefficients and the corresponding decoded coefficients reconstructed directly by inverse transform and quantization, which can be regarded as a data fidelity constraint. The second term, $\mathcal{D}_2$, constrains the similarity between original coefficients and their predictions acquired from a temporal smooth model, temporal auto-regressive (TAR) model used in this paper, which is



(a) (b)

130	120	127	132
125	119	123	123
121	124	116	128
132	130	129	140

(c)

389	289	193	108
215	159	122	67
111	85	73	46
59	44	37	21

(d)

Fig. 3. Compression noise distribution in 4×4 blocks in spatial domain and DCT domain. (a) The histogram of compression noise in spatial domain, (b) the histogram of compression noise in DCT domain, (c) the variance of compression noise in spatial domain, (d) the variance of compression noise in DCT domain.

regarded as a video temporal prior model. The third term, $\mathcal{D}_3$, measures the conformance between original coefficients and the weighted average of non-local similar blocks, which are acquired from both current frame and neighboring frames. Due to the variation of video content, only one image prior mode is difficult to describe image contents with different statistical characteristics. The estimation from multiple hypotheses may achieve better results when assigning higher weights to the model, which describes target image structure more accurate. Therefore, the three distance metrics are normalized by their estimation reliabilities. In the following subsections, each of the estimations and its corresponding reliability are introduced in detail.

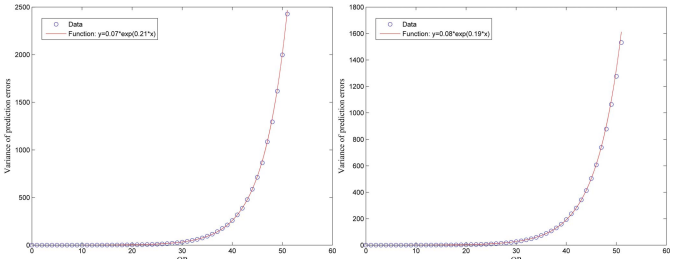
B. Data Fidelity Constraint

In the proposed method, we utilize the reconstructed coefficients, $\mathbf{Y}_{\mathcal{B}_t}(u, v)$, in every $N \times N$ block directly derived from the decoded image as data fidelity constraint, and call these coefficients *decoded prediction*. Since the errors of *decoded prediction* are mainly caused by quantization, we take the variance of quantization noise to reflect its reliability. Therefore, the first distance term is formulated as follows,

$$\mathcal{D}_1(\mathbf{X}_{\mathcal{B}_t}, \mathbf{Y}_{\mathcal{B}_t}) = \sum_{u,v=0}^{N} \frac{(\mathbf{X}_{\mathcal{B}_t}(u, v) - \mathbf{Y}_{\mathcal{B}_t}(u, v))^2}{\sigma^2_{QP}(u, v)}, \quad (3)$$

where $\sigma^2_{QP}(u, v)$ is the variance of the compression noise for band (u, v), and its reciprocal is utilized as the reliability of *decoded prediction*.

In Eqn.(3), the distance is measured in DCT domain rather than in spatial domain mainly based on two reasons. Firstly, the compression noises in spatial domain are correlated and dominant by DC components. This makes it difficult to estimate the original coefficients in other bands, which represent image structures, e.g., edges or textures. Fig.3(a) and Fig.3(c) illustrate the histogram and variance of compression noise in



(a) (b)

Fig. 4. Relationship between the variance of compression noise and *QP*, (a) band (0, 0), (b) band (0, 1).

4×4 blocks of spatial domain. For different positions in spatial domain, compression noise almost follows the same distribution with approximate noise variances. However, in transform domain, compression noise shows quite different distributions in different bands, e.g., in Fig.3(c) and Fig.3(d). The noise variance decreases from low frequency bands to high frequency bands obviously. The good statistical characteristic of compression noise in transform domain is useful to distinguish reliability of *decoded prediction*.

In addition, since compression noise is mainly caused by quantization of coefficients, its variance is easy to estimate in transform domain. Based on statistical results, we take exponential function, $y = ae^{bx}$, to fit the variance of compression noise with QP, which can be derived from compression stream directly. Here, y represents the noise variance and x represents QP. Fig. 4 shows two examples of the relationship between the variance of compression noise and QP for bands (0, 0) and (0, 1), where the exponential functions well fit variation between the variance of compression noise and QP. Therefore, we can easily learned exponential function parameters (a, b) for compression noise in all the bands of transform-block.

C. Temporal Auto-Regressive Estimation

Auto-regressive model is widely used to represent image local structures for images or videos. In [27] and [28], the researchers employ the AR model to interpolate high resolution image. In [29], the authors utilize AR model to represent the temporal relationship of neighboring video frames in frame rate up conversion. However, these methods apply AR model in pixel domain as follows.

$$\boldsymbol{y}_t = \sum_{i=0}^{P} a_i \boldsymbol{x}_{t-i} + \boldsymbol{\epsilon}_t, \quad (4)$$

where a_i is the model parameter and $\boldsymbol{\epsilon}_t$ is the model error. These parameters can be derived via least square methods. In our proposed method, we take the temporal auto-regressive (TAR) model in transform domain to represent the relationship of blocks in different frames along their trajectories. Based on the statistical results in TABLE I, the temporal correlations among temporal blocks dominate by DC component, which makes only one AR model difficult to well represent the content in blocks. Therefore, in this paper, we take two TAR models to represent the relationship of DC and AC components among temporal transform blocks, respectively.

TABLE I

TEMPORAL CORRELATION COEFFICIENTS (CC) IN 4 × 4 BLOCKS ALONG TRAJECTORIES IN DIFFERENT DOMAIN

CC in pixel domain				CC in transform domain			
0.988	0.988	0.989	0.988	0.998	0.962	0.913	0.832
0.989	0.988	0.989	0.988	0.948	0.899	0.876	0.790
0.989	0.987	0.989	0.988	0.885	0.843	0.790	0.753
0.987	0.988	0.989	0.987	0.803	0.768	0.709	0.590

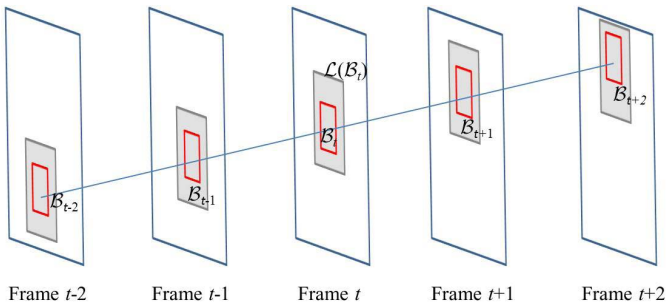


Fig. 5. TAR model parameter estimation from neighboring transform blocks along trajectories.

For one block $\mathcal{B}_t$ in Fig.5, we first find its corresponding blocks from forward and backward frames along its trajectory. Then, the prediction generated by $\{\mathbf{Y}_{\mathcal{B}_k}\}$ with TAR in Eqn.(5) is called *temporal prediction*, and formulated as,

$$\mathbf{Y}_{AR(\mathcal{B}_t)}(u,v) = \sum_{\substack{i=-p \\ i\neq 0}}^{p} \alpha_i \mathbf{Y}_{\mathcal{B}_{t+i}}(u,v) \tag{5}$$

The model parameters, $\{\alpha_i\}$, are estimated from neighboring transform blocks in a cube regions, i.e., blocks in the gray area in Fig.5, by solving optimization problem in Eqn.(6).

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \parallel \mathbf{Y}^m_{\mathcal{B}_t} - \sum_{\substack{i=-p \\ i\neq 0}}^{p} \alpha_i \mathbf{Y}_{\mathcal{B}_{t+i}} \parallel_2^2 \tag{6}$$

where m represents DC and AC respectively, and $\boldsymbol{\alpha}$ is the vector composed of $\{\alpha_i\}$. $\mathbf{Y}^m_{\mathcal{B}_t}$ is a vector composed of DC/AC component(s) of transform blocks in a neighborhood $\mathcal{L}(\mathcal{B}_t)$ of t^{th} frame.

Obviously, for different regions and times, videos show different temporal correlations, which leads to the TAR predictions in Eqn.(5) with different reliabilities. In this paper, we take variance of the local prediction errors to reflect prediction reliability of TAR. The prediction errors for the local area, $\mathcal{L}(\mathcal{B}_t)$, is,

$$\boldsymbol{e}_{\mathcal{B}'_t} = \mathbf{Y}_{\mathcal{B}'_t} - \sum_{\substack{i=-p \\ i\neq 0}}^{p} \alpha_i \mathbf{Y}_{\mathcal{B}'_{t+i}}, \quad \mathcal{B}'_t \in \mathcal{L}(\mathcal{B}_t) \tag{7}$$

And its variance is calculated with the following equations,

$$\mu_e(u,v) = \frac{1}{M} \sum_{\mathcal{B}'_t \in \mathcal{L}(\mathcal{B}_t)} \boldsymbol{e}_{\mathcal{B}'_t}(u,v) \tag{8}$$

$$\sigma_e^2(u,v) = \frac{1}{M} \sum_{\mathcal{B}'_t \in \mathcal{L}(\mathcal{B}_t)} \left(\boldsymbol{e}_{\mathcal{B}'_t}(u,v) - \mu_e(u,v) \right)^2, \quad \mathcal{B}'_t \in \mathcal{L}(\mathcal{B}_t) \tag{9}$$

where $M = |\mathcal{B}_t|$ represents the number of the blocks in the neighborhood $\mathcal{L}(\mathcal{B}_t)$. Here, $\sigma_e^2(u,v)$ is the variance of TAR model errors in band (u,v) to predict block $\mathcal{B}_t$. Due to the TAR parameters derived based on decoded frames, which are distorted by quantization noise, we add a scaled variance of compressed noise to reflect the prediction reliability of TAR model to predict the original coefficients.

$$\sigma^2_{AR}(u,v) = \sigma_e^2(u,v) + s_1 \sigma^2_{QP}(u,v) \tag{10}$$

where s_1 is a scale parameter.

Therefore, the second distance metric, $\mathcal{D}_2$, in Eqn.(2) is defined as follows,

$$\mathcal{D}_2\left(\mathbf{X}_{\mathcal{B}_t}, \left\{\mathbf{Y}_{\mathcal{B}_k}\right\}\right) = \sum_{u,v=0}^{N} \frac{\left(\mathbf{X}_{\mathcal{B}_t}(u,v) - \mathbf{Y}_{AR(\mathcal{B}_t)}(u,v)\right)^2}{\sigma^2_{AR(\mathcal{B}_t)}(u,v)}, \tag{11}$$

Here $1/\sigma^2_{AR(\mathcal{B}_t)}(u,v)$ is utilized as the *temporal prediction* reliability of band (u,v) for block $\mathcal{B}_t$.

D. Spatio-Temporal Non-Local Estimation

Considering the complexity of video contents, which are difficult to represent only with the TAR model, e.g., occlusion areas and fast motion objects, we further utilize non-local similar blocks to estimate original coefficients in this paper. For one block, $\mathcal{B}_t$, we select K-nearest neighbors from current frame and neighboring frames based on L2 norm of difference of transform blocks, and take the weighted average of these blocks as *non-local prediction* in Eqn.(12).

$$\mathbf{Y}_{\mathcal{N}(\mathcal{B}_t)}(u,v) = \omega_i \mathbf{Y}_{\mathcal{B}_i}(u,v), \quad \mathcal{B}_i \in \mathcal{N}(\mathcal{B}_t) \tag{12}$$

$$\omega_i = \frac{1}{Z} exp\left(-\frac{d_i}{h}\right) \tag{13}$$

$$d_i = \sum_{u,v=0}^{N} \left(\mathbf{Y}_{\mathcal{B}_i}(u,v) - \mathbf{Y}_{\mathcal{B}_t}(u,v)\right)^2 \tag{14}$$

Here h is a smoothness parameter, Z is the normalization factor and $\mathcal{N}(\mathcal{B}_t)$ is the set of non-local similar blocks.

Since the *non-local prediction* is based on image self-similarity with the assumption that these similar neighbors following the identical distribution, when the selected K-nearest neighbors are more similar with the target one, the non-local prediction is more accurate. On the contrary, if the K-nearest neighbors are not similar with the target one, the non-local prediction may generate large errors. The variance of coefficients in K-nearest neighbors can reflect their similarity to the target one, and further to describe the reliability of the non-local prediction. The variance of non-local coefficients is estimated based on the decoded frames as follows,

$$\sigma^2_{\mathcal{N}_{\mathbf{Y}}(\mathcal{B}_t)}(u,v) = \sum_{i=1}^{K} \omega_i \left(\mathbf{Y}_{\mathcal{B}_i}(u,v) - \mathbf{Y}_{\mathcal{N}(\mathcal{B}_t)}(u,v)\right)^2 \tag{15}$$

Considering the existence of compression noise, we revise the variance by adding the scaled variance of compression

noise to make it better reflect the reliability of *non-local prediction* for the original coefficients.

$$\sigma^2_{\mathcal{N}(\mathcal{B}_t)}(u,v) = \sigma^2_{\mathcal{N}_{\mathbf{Y}}(\mathcal{B}_t)}(u,v) + s_2\sigma^2_{QP}(u,v) \tag{16}$$

Based on the *non-local prediction*, the third distance metric, $\mathcal{D}_3$, is defined as,

$$\mathcal{D}_3\left(\mathbf{X}_{\mathcal{B}_t}, \left\{\mathbf{Y}_{\mathcal{B}_k}\right\}\right) = \sum_{u,v=0}^{N} \frac{\left(\mathbf{X}_{\mathcal{B}_t}(u,v) - \mathbf{Y}_{\mathcal{N}(\mathcal{B}_t)}(u,v)\right)^2}{\sigma^2_{\mathcal{N}(\mathcal{B}_t)}(u,v)} \tag{17}$$

IV. Optimization Solution

In Eqn.(2), the optimization problem is related to all the possible blocks in an image, which are overlapped and dependent. In order to solve the problems, we divide these overlapped blocks in set Ω into several subsets $\Omega^{sub}_{i,j}$, where the blocks are non-overlapped.

$$\Omega = \left\{\mathcal{B}_{m,n} | 0 \le m \le H - N, 0 \le n \le W - N\right\} \tag{18}$$

$$\Omega^{sub}_{i,j} = \left\{\mathcal{B}_{i,j} | (i \bmod N) \equiv 0, (j \bmod N) \equiv 0\right\} \tag{19}$$

where $\mathcal{B}_{m,n}$ is a block in frame $\boldsymbol{x}_t$ with its top-left pixel being $\boldsymbol{x}_t(m,n)$. There are a total of $N \times N$ subsets, where N is the transform size ($N = 8$ in our experiments). Each subset forms a complete coverage of target frame, $\boldsymbol{x}_t$, with non-overlapped blocks except at image boundaries. Therefore, the solution to the optimization problem in Eqn.(2) is derived by solving $N \times N$ sub-optimization problems to minimize Eqn.(2) w.r.t the variable $\mathbf{X}_{\mathcal{B}_t}$ in a block subset $\Omega^{sub}_{i,j}$ while keeping the estimated $\mathbf{X}_{\mathcal{B}_t}$ in other block subsets temporarily constant and irrelevant. By setting the deviation of Eqn.(2) to zero, the solution for each band is derived as follows,

$$\hat{\mathbf{X}}_{\mathcal{B}_t}(u,v) = c_1\mathbf{Y}_{\mathcal{B}_t}(u,v) + c_2\mathbf{Y}_{AR(\mathcal{B}_t)}(u,v) + c_3\mathbf{Y}_{\mathcal{N}(\mathcal{B}_t)}(u,v) \tag{20}$$

$$\begin{cases} \eta_1 = \sigma^2_{QP}(u,v)\sigma^2_{AR(\mathcal{B}_t)}(u,v) \\ \eta_2 = \sigma^2_{QP}(u,v)\sigma^2_{\mathcal{N}(\mathcal{B}_t)}(u,v) \\ \eta_3 = \sigma^2_{AR(\mathcal{B}_t)}(u,v)\sigma^2_{\mathcal{N}(\mathcal{B}_t)}(u,v) \end{cases} \tag{21}$$

$$c_i = \eta_i / \sum_{i=1}^{3} \eta_i \tag{22}$$

Due to the fact that the blocks are overlapped, every pixel has multiple estimations from different blocks. We aggregate all these estimations adaptively to generate the final high quality videos. For every block, we take the weighted average of its prediction variance, $\{\sigma^2_{QP}, \sigma^2_{AR(\mathcal{B}_t)}, \sigma^2_{\mathcal{N}(\mathcal{B}_t)}\}$, to calculate its weight used in overlapped block fusion, which is formulated as follows,

$$\omega_{\mathcal{B}_t} = \frac{1}{\sum_{u,v=0}^{N}\sum_{i=1}^{3} c_i\sigma_i^2(u,v)} \tag{23}$$

where c_i is calculated according to Eqn.(22) and σ^2_{QP}, $\sigma^2_{AR(\mathcal{B}_t)}$ and $\sigma^2_{\mathcal{N}(\mathcal{B}_t)}$ are denoted as σ_1^2, σ_2^2 and σ_3^2, respectively.

Algorithm 1 The Proposed Algorithm

Input: Compressed video frames, $\{\boldsymbol{y}_t\}$.
for *each frame* $\boldsymbol{y}_t$ **do**
  Parameter initialization, $\sigma^2_{QP}(u,v), s_1, s_2, h$;
  for *each subset* $\Omega^{sub}_{i,j}$ **do**
    Calculate TAR prediction and its variance via Eqn.(5), Eqn.(9) and Eqn.(10);
    Calculate nonlocal prediction and its variance via Eqn.(12), Eqn.(15) and Eqn.(16);
    Multiple predictions adaptive fusion via Eqn.(20)-(22);
    Calculate weights for overlapped blocks via Eqn.(23);
  end
  Adaptive fusion of overlapped blocks according to Eqn.(24);
  Divide the estimated frame into blocks according to TU partition, and apply QC in Eqn.(26) to them;
  Inversely transform image into spatial domain,$\{\boldsymbol{x}''_t\}$;
end
Output: High quality video frames, $\{\boldsymbol{x}''_t\}$

Therefore, different predictions are weighted and averaged to generate reconstruction frame, $\hat{\boldsymbol{x}}'$.

$$\hat{\boldsymbol{x}}'(i,j) = \frac{\sum_{\mathcal{B}'\in S_{i,j}} \omega_{\mathcal{B}'}\hat{\boldsymbol{x}}_{\mathcal{B}'}(i,j)}{\sum_{\mathcal{B}'\in S_{i,j}} \omega_{\mathcal{B}'}} \tag{24}$$

$$S_{i,j} = \{\mathcal{B} | \boldsymbol{x}(i,j) \in \mathcal{B}\} \tag{25}$$

where $\hat{\boldsymbol{x}}_{\mathcal{B}'}(i,j)$ is the estimated pixel value in block $\mathcal{B}'$.

Considering the original transform coefficients being in $\left[\mathbf{Y}_{\mathcal{B}_t}(u,v) - Q/2,\ \mathbf{Y}_{\mathcal{B}_t}(u,v) + Q/2\right]$ (Q is the quantization step), we further divided the reconstructed frames into blocks the same as transform unit (TU) division in coding process, which can be derived from the compression stream, and then utilize a projection onto convex set operation, $x' = \mathcal{P}_Q(x,y)$, to constrain coefficients into quantization intervals. This operation in Eqn.(26) is denoted as quantization constraint (QC) in this paper.

$$\hat{\mathbf{X}}''_{\mathcal{B}_t}(u,v) = \begin{cases} \mathbf{Y}_{\mathcal{B}_t}(u,v) + \frac{Q}{2}, & if\ \hat{\mathbf{X}}'_{\mathcal{B}_t}(u,v) > \mathbf{Y}_{\mathcal{B}_t}(u,v) + \frac{Q}{2} \\ \mathbf{Y}_{\mathcal{B}_t}(u,v) - \frac{Q}{2}, & if\ \hat{\mathbf{X}}'_{\mathcal{B}_t}(u,v) < \mathbf{Y}_{\mathcal{B}_t}(u,v) - \frac{Q}{2} \\ \hat{\mathbf{X}}'_{\mathcal{B}_t}(u,v), & others \end{cases} \tag{26}$$

Therefore, based on the above discussion, our proposed algorithm is described in Algorithm 1.

There are some significant differences between our proposed method and that in [19]. First of all, we take advantage of transform domain TAR model to describe the relationship of video signals along their trajectories, and utilize different parameters to model DC and AC components respectively, which is different from traditional AR model in spatial domain. Based on the following experimental results, the temporal extension with TAR model is very efficient for video quality improvement. Second, in non-local similar block estimation,

we select the K-nearest neighbors instead of all the non-local blocks to construct prediction, which excludes some outliers directly. Third, in overlapped aggregation stage illustrated in Eqn.(23) and Eqn.(24), we take the weighted average of every pixel based on their prediction reliabilities instead of average in [19]. In addition, we also modify the variance estimation of non-local estimation with scaled compression noise, i.e., Eqn.(16), which makes it more reasonable theoretically.

V. Experimental Results

In this paper, we evaluate the proposed method based on HEVC compressed videos, and compare its performance with state-of-the-art compression artifact reduction algorithms and denoising algorithms, including in-loop filter of HEVC (deblocking filter and SAO) [3], [30], KSVD [16], BM3D [17], CSR [21], LPG-PCA [20], the non-local means filter (NLM) [9], FoE method [31], and Zhang's method [19]. The HEVC reference software is HM12.0. For the compared methods, we try out many parameters with some test images to find the reasonable ones under different compression ratios.

The parameters used in our proposed method are configured as follows. First, we set the block size $N = 8$ and the number of nearest neighbors $K = 50$, which are empirical values. For the compression noise variance, it should be adaptive according to QP and coding modes (intra coding and inter coding). We first find $\sigma^2_{QP}(u, v)$ from HEVC compressed video sequences, *BassketballPass* and *BlowingBubbles*, with QP from 0 to 51 for intra coding and inter coding modes respectively. Then, exponential functions are utilized to model the relationship between $\sigma^2_{QP}(u, v)$ and QP just as that in Fig.4. The parameter, p, is the temporal radius for TAR model, which may cause higher computation complexity and larger delay for real time applications. We set $p = 2$ in our experiments for all the videos as a trade-off between complexity, delay and performance. Fig.6 illustrates the performance variation with p, and the performance of our method can achieve more improvements when using more temporal frames. Based on the assumption that the compression noises in different blocks are independent in transform domain, then the variance of compression noise in weighted average blocks should be divided by the number of blocks, i.e., $s_1 = 1/2p$ (p is the temporal radius in Eqn.(7)) and $s_2 = 1/K$. Fortunately, although they may not be the optimal values, the performance of our method is not sensitive to the two parameters s_1 and s_2. Finally, smoothness factor, h, in Eqn.(13) should be related with compression noise as suggested in [9], which is difficult to determine. In our method, we find its suitable values according to the reconstruction performance for compressed video sequences, *BassketballPass* and *BlowingBubbles*, with different QPs and coding modes. In this paper, we take $h = 5$ and $h = 30$ when QP is equal to 27 for intra and inter coding modes, respectively. Although these parameters are derived according to the reconstruction performance of the proposed method from a few compressed video sequences with different QPs and coding modes, they are useful for most of the compressed videos based on the experimental results.

We take some common test sequences with CIF and WQVGA formats, which are widely used in video coding.

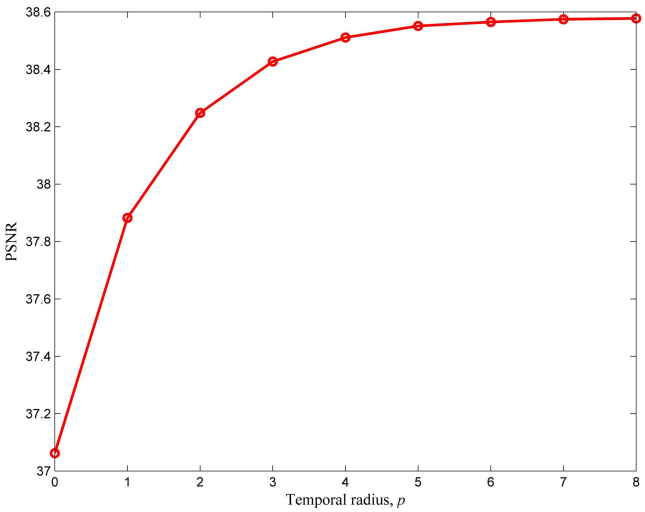


Fig. 6. The PSNR results of our method with different temporal radius p for the 8th frame in sequences, *CITY*, compressed by HEVC (AI) at QP=27.

These sequences are first compressed by HEVC with two coding configurations, i.e., all intra (AI) coding and low delay P coding (LDP). Then different compression artifact reduction methods are applied to the luminance component of these compressed sequences without in-loop filter to restore high quality videos. The average PSNR results of the reconstructed 10 frames in each sequence are listed in TABLE II and IV. Herein, TABLE II show the results for compressed sequences with AI configuration, and TABLE IV shows the results for compressed sequences with LDP configuration. The quality of HEVC reconstruction without in-loop filter is denoted as HEVC-N, and the quality of HEVC reconstruction with in-loop filter is denoted as HEVC-I. Based on the results, our proposed scheme outperforms all of the other methods and achieves up to 1.14 dB gain over HEVC decoder without in-loop filters on average for AI configuration. Compared with other artifact reduction methods, our proposed method also achieves about 0.76∼1.19 dB on average at QP=27 with AI configuration. Especially, our method achieves up to 1.50 dB gains over HEVC-N for sequences *Flowervase*. Compared with our previous work, Zhang's method [19], the proposed method also achieves significant gain especially for AI coding configuration, which verifies the efficiency of our proposed TAR model and weighted aggregation. Since the PSNR quality metric is not well consistent with perceptual quality, we also take the widely used quality metric, Structural Similarity Index Metric (SSIM) [33], to further verify the performance of our proposed method. TABLE III and V list the corresponding SSIM results for reconstructed videos from compressed ones with AI and LDP configurations, respectively. Based on SSIM results, our proposed method also outperforms the comparison methods, and achieves about 0.005∼0.008 and 0.004∼0.001 SSIM improvement on average for AI and LDP configurations, respectively. Fig.7 shows the overall performance for all the sequences compressed at QP=27. Fig.8 shows the performance of different reconstruction methods for every frame in sequences *Students* and *Flowervase*, and our method achieves significant improvement

TABLE II
THE AVERAGE PSNR RESULTS WITH DIFFERENT METHODS, SEQUENCES COMPRESSED BY HEVC AI CODING AT QP=27 (UNIT: dB)

Sequences	HEVC-N	HEVC-I	SKR	NLM	KSVD	SA-DCT	BM3D	FoE	CSR	LPG-PCA	Zhang's	Proposed
Bus	37.35	37.44	37.44	37.35	37.37	37.58	37.64	37.55	37.71	37.69	37.58	38.48
City	37.13	37.21	36.63	37.13	37.14	37.38	37.40	37.28	37.48	37.46	37.33	38.57
Foreman	38.74	38.92	38.75	38.74	38.75	38.88	38.95	38.91	39.19	39.04	38.97	39.62
Students	38.63	38.77	38.76	38.63	38.64	38.86	38.89	38.81	39.02	38.98	38.82	40.11
News	40.07	40.25	40.26	40.06	40.09	40.47	40.59	40.40	40.62	40.66	40.43	41.49
Carphone	39.95	40.19	39.10	39.95	39.97	40.35	40.46	40.23	40.51	40.52	40.30	40.94
Mobile	37.01	37.08	37.10	37.01	37.05	37.16	37.31	37.17	37.35	37.36	37.17	38.18
Soccer	37.13	37.20	36.88	37.13	37.14	37.34	37.26	37.26	37.35	37.41	37.32	38.25
Stefan	38.51	38.64	38.61	38.51	38.53	38.79	38.87	38.78	38.93	38.92	38.80	39.35
BasketballPass	38.96	39.10	39.03	38.96	38.99	39.23	39.28	39.16	39.36	39.31	39.17	40.00
BlowingBubbles	36.90	36.99	36.84	36.89	36.90	37.05	37.02	36.96	37.12	37.11	37.02	37.98
BQSquare	37.07	37.13	36.88	37.07	37.07	37.03	37.03	37.05	37.21	37.15	37.02	38.20
Flowervase	43.48	43.66	43.67	43.47	43.50	43.92	43.96	43.84	44.00	43.92	43.84	44.98
Keiba	38.38	38.55	38.57	38.38	38.40	38.72	38.67	38.68	38.76	38.76	38.66	39.26
RaceHorses	37.96	38.12	38.12	37.96	37.97	38.31	38.34	38.21	38.41	38.43	38.28	39.06
Average	38.49	38.62	38.44	38.48	38.50	38.74	38.78	38.69	38.87	38.85	38.71	39.63

TABLE III
THE AVERAGE SSIM RESULTS WITH DIFFERENT METHODS, SEQUENCES COMPRESSED BY HEVC AI CODING AT QP=27

Sequences	HEVC-N	HEVC-I	SKR	NLM	KSVD	SA-DCT	BM3D	FoE	CSR	LPG-PCA	Zhang's	Proposed
Bus	0.964	0.964	0.964	0.964	0.964	0.965	0.964	0.965	0.965	0.965	0.965	0.971
City	0.958	0.959	0.952	0.958	0.958	0.960	0.960	0.959	0.961	0.961	0.960	0.969
Foreman	0.949	0.950	0.948	0.949	0.949	0.948	0.948	0.950	0.951	0.949	0.950	0.954
Students	0.958	0.959	0.959	0.958	0.958	0.960	0.959	0.959	0.961	0.961	0.960	0.967
News	0.971	0.973	0.973	0.971	0.971	0.974	0.974	0.974	0.974	0.974	0.973	0.976
Carphone	0.963	0.965	0.964	0.963	0.963	0.965	0.966	0.965	0.966	0.966	0.965	0.968
Mobile	0.978	0.978	0.979	0.978	0.978	0.979	0.980	0.979	0.980	0.980	0.979	0.983
Soccer	0.944	0.944	0.938	0.944	0.944	0.946	0.942	0.944	0.944	0.946	0.946	0.955
Stefan	0.982	0.982	0.982	0.982	0.982	0.983	0.983	0.983	0.983	0.984	0.983	0.985
BasketballPass	0.950	0.951	0.950	0.950	0.950	0.952	0.952	0.952	0.953	0.953	0.952	0.958
BlowingBubbles	0.943	0.944	0.942	0.943	0.944	0.944	0.943	0.943	0.945	0.945	0.945	0.953
BQSquare	0.944	0.944	0.937	0.944	0.944	0.940	0.939	0.942	0.943	0.943	0.943	0.950
Flowervase	0.982	0.983	0.983	0.982	0.982	0.985	0.984	0.984	0.984	0.985	0.984	0.986
Keiba	0.963	0.965	0.965	0.963	0.964	0.966	0.965	0.965	0.965	0.966	0.965	0.969
RaceHorses	0.963	0.965	0.965	0.963	0.963	0.966	0.966	0.965	0.966	0.967	0.966	0.971
Average	0.961	0.962	0.960	0.961	0.961	0.962	0.962	0.962	0.963	0.963	0.962	0.968

for every frame. Fig.9 shows the reconstruction quality of different methods in a large QP range, which corresponds to a large bitrate range. We can see that our proposed scheme works well over a wide bitrate range and achieves better quality than other methods in different QPs. Especially, for middle and high bitrate, the performance of our proposed method also achieves obvious improvement than that of others due to the decoded coefficients with higher reliability and quantization constraint, which prevent from smoothing image textures excessively. For LDP configuration, the gain of our method is not so significant as that for AI configuration. This is because the high correlation of compression noise in temporal domain makes the performance of the TAR model degraded.

Fig.10 and Fig.11 illustrate the subjective quality of the reconstructed images which are compressed at QP=37. From the subjective quality comparison, we can see that the compression artifacts are obvious in the images reconstructed by the standard HEVC decoder without in-loop filters. The compared methods are able to reduce the compression artifacts partially, but also make the image blurring around edges. Our proposed method produces more pleasing visual quality than that of other methods. It does not only reduce most of the compression artifacts significantly, but also preserves image edges very well.

The computation required by the proposed algorithm mainly resides in the derivation process for temporal prediction and

TABLE IV

THE AVERAGE PSNR RESULTS WITH DIFFERENT METHODS, SEQUENCES COMPRESSED BY HEVC LDP CODING AT QP=27 (UNIT: dB)

Sequences	HEVC-N	HEVC-I	SKR	NLM	KSVD	SA-DCT	BM3D	FoE	CSR	LPG-PCA	Zhang's	Proposed
Bus	35.86	36.06	36.00	36.01	36.02	36.14	36.18	36.12	36.24	36.24	36.21	36.37
City	36.17	36.24	35.66	36.23	36.23	36.24	36.17	36.12	36.29	36.25	36.24	36.33
Foreman	37.85	38.27	38.11	38.16	38.17	38.12	38.15	38.14	38.38	38.24	38.26	38.40
Students	38.05	38.28	38.20	38.26	38.26	38.28	38.24	38.20	38.40	38.35	38.32	38.44
News	39.25	39.48	39.53	39.45	39.47	39.62	39.74	39.62	39.79	39.80	39.64	39.75
Carphone	39.01	39.43	39.13	39.34	39.35	39.51	39.64	39.44	39.70	39.66	39.56	39.66
Mobile	35.22	35.35	35.35	35.30	35.33	35.39	35.50	35.43	35.53	35.53	35.44	35.83
Soccer	36.09	36.23	35.84	36.20	36.20	36.26	36.13	36.16	36.26	36.33	36.31	36.34
Stefan	36.69	36.97	37.00	36.89	36.90	37.10	37.27	37.19	37.27	37.30	37.25	37.32
BasketballPass	38.21	38.44	38.32	38.39	38.40	38.47	38.51	38.40	38.59	38.53	38.49	38.62
BlowingBubbles	35.59	35.76	35.60	35.72	35.73	35.74	35.67	35.65	35.79	35.77	35.76	35.98
BQSquare	35.64	35.76	35.53	35.71	35.70	35.63	35.61	35.63	35.78	35.70	35.66	36.30
Flowervase	42.87	43.11	42.99	43.08	43.09	43.11	43.11	43.03	43.20	43.03	43.14	43.33
Keiba	37.30	37.51	37.45	37.45	37.46	37.53	37.44	37.48	37.54	37.56	37.55	37.50
RaceHorses	36.36	36.67	36.64	36.56	36.57	36.79	36.80	36.73	36.86	36.90	36.83	36.95
Average	37.34	37.57	37.42	37.52	37.52	37.60	37.61	37.56	37.71	37.68	37.64	37.81

TABLE V

THE AVERAGE SSIM RESULTS WITH DIFFERENT METHODS, SEQUENCES COMPRESSED BY HEVC LDP CODING AT QP=27

Sequences	HEVC-N	HEVC-I	SKR	NLM	KSVD	SA-DCT	BM3D	FoE	CSR	LPG-PCA	Zhang's	Proposed
Bus	0.956	0.957	0.956	0.957	0.957	0.957	0.955	0.956	0.957	0.957	0.958	0.959
City	0.952	0.952	0.943	0.952	0.952	0.952	0.950	0.949	0.951	0.952	0.952	0.953
Foreman	0.943	0.947	0.943	0.946	0.946	0.942	0.942	0.944	0.946	0.944	0.945	0.947
Students	0.955	0.958	0.956	0.957	0.958	0.956	0.955	0.955	0.957	0.957	0.957	0.958
News	0.968	0.971	0.971	0.971	0.971	0.972	0.972	0.971	0.972	0.972	0.972	0.972
Carphone	0.959	0.962	0.962	0.962	0.962	0.962	0.963	0.962	0.963	0.963	0.963	0.963
Mobile	0.974	0.975	0.975	0.975	0.975	0.975	0.976	0.975	0.976	0.976	0.975	0.977
Soccer	0.937	0.937	0.926	0.937	0.937	0.935	0.929	0.931	0.933	0.935	0.936	0.936
Stefan	0.976	0.978	0.979	0.978	0.978	0.979	0.980	0.979	0.980	0.980	0.979	0.980
BasketballPass	0.947	0.949	0.946	0.949	0.949	0.948	0.948	0.947	0.949	0.949	0.949	0.950
BlowingBubbles	0.933	0.935	0.932	0.935	0.935	0.933	0.931	0.931	0.934	0.934	0.934	0.937
BQSquare	0.944	0.943	0.934	0.943	0.943	0.937	0.935	0.937	0.941	0.940	0.941	0.945
Flowervase	0.982	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.984	0.984
Keiba	0.957	0.959	0.958	0.958	0.958	0.958	0.956	0.957	0.958	0.958	0.958	0.958
RaceHorses	0.950	0.954	0.953	0.953	0.953	0.955	0.954	0.954	0.955	0.956	0.955	0.956
Average	0.956	0.957	0.954	0.957	0.957	0.956	0.955	0.955	0.957	0.957	0.957	0.958

non-local prediction. In the process to generate temporal prediction, motion estimation and temporal auto-regressive (TAR) model estimation are two main modules that require intensive computation. In our implementation, exhaustive full search is employed to find the best motion vector for every $N \times N$ block in a $R \times R$ search range of every reference frame. It needs about $2pN^2R^2$ subtractions and additions, and $log(2pR^2)$ comparisons for every block, where $2p$ is the number of reference frames. In the process to solve the TAR model parameters via least square optimization, the computation cost mainly comes from three matrix multiplication operations and one matrix inversion operation. This needs about $\{(2p)^2N^2+2pN^2+C(2p)^3\}$ multiplications, where C is a constant. In the process to generate non-local prediction, the computation mainly comes from the retrieval of K-nearest neighbours. For every block, a naive brute force search needs about $2pN^2R^2$ subtractions and additions, and $Klog(2pR^2)$ comparisons with Heap's algorithm to find the K-nearest neighbours. These computations are also required by other non-local based methods, such as NLM, BM3D, CSR and LPG-PCA, etc. Since the amount of computation for each block is a constant in our algorithm, the overall complexity increases with the image dimension only linearly, i.e., the order of overall complexity is $O(kHW)$, where k is a constant, W and H are image width and height respectively.

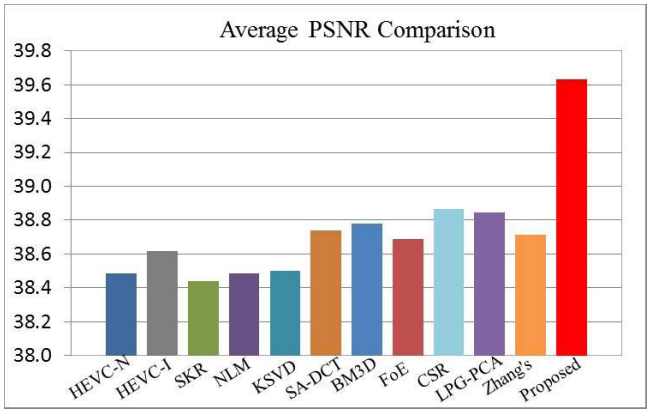


Fig. 7. Average PSNR comparison for all the sequences compressed by HEVC (AI) at QP=27.

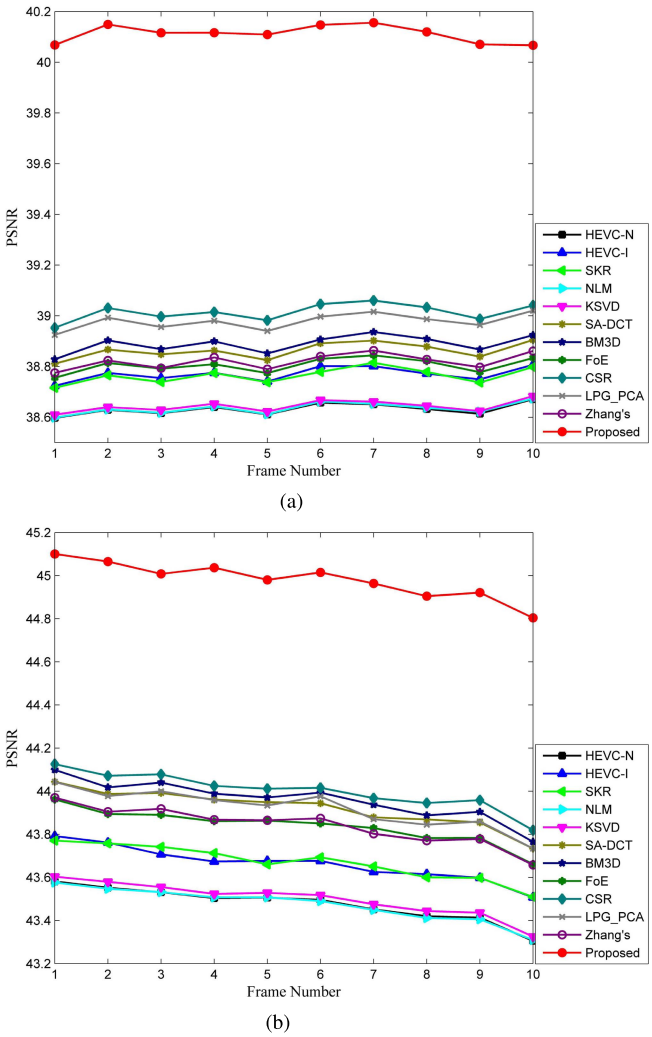


Fig. 8. Performance of different reconstruction methods for every frame of sequence compressed by HEVC (AI). (a) *Students*. (b) *Flowervase*.

At present, our algorithm is implemented with MATLAB and C++ without parallel optimization. In order to give readers an intuitive impression of complexity and efficiency of the proposed method, we test the running time of different methods with MATLAB 2012, Intel (R) Core (TM) i5-4570@3.20GHz, and 64bit Window 7 operating system.

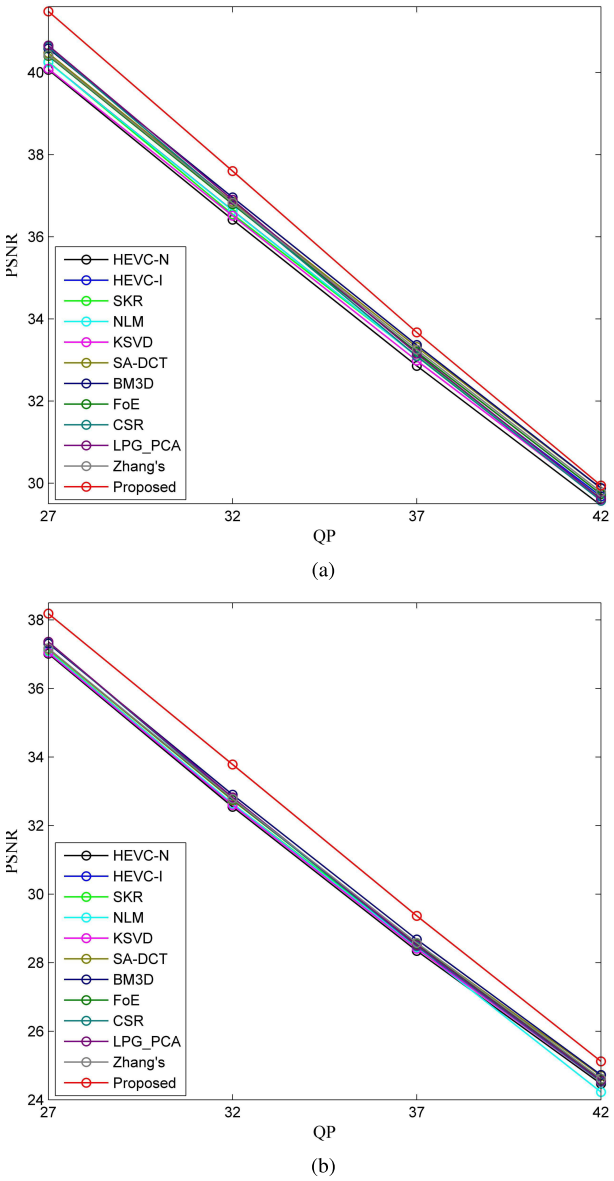


Fig. 9. Average quality of the reconstructed video sequences with different methods at different bitrates. (a) *News*. (b) *Mobile*.

Since our method need to solve different non-overlapped block sets, its complexity is dependent on the number of processed block sets. Fig.12 shows the relationship between complexity and performance of our algorithm, where the horizontal axis represents both running time and number of processed block sets, and the vertical axis represents the PSNR values of reconstructed frames in sequence, *Students*. The computation complexity increases with the amount of subsets of blocks in Eqn.(19), while the performance is also improved further. There are about 0.2dB improvement compared with non-overlapped process. In practice, users can also adjust the number of processed block sets according their requirement for computation complexity. In our experiments, we reconstruct high quality videos by processing 16 block sets as a trade-off between complexity and efficiency. Fig.13 shows the average running time for these methods, most of which are also implemented with MATLAB and C++, except

Fig. 10. Reconstructed frames with different methods. The test video sequence, *Stefan*, is compressed by HEVC (AI) at QP=37. (a) HEVC-N. (b) HEVC-I. (c) SKR. (d) NLM. (e) SKR. (f) SA-DCT. (g) BM3D. (h) FoE. (i) CSR. (j) LPG-PCA. (k) Zhang's. (l) Proposed.

for SKR, FoE, CSR and LPG-PCA. The methods, SA-DCT and BM3D, are very fast because they are implemented with well optimized codes. Although our method needs about 13s for a CIF/WQVAG frame on average, it has the closed-form solution for every block as formulated in Eqn.(20), which means that it could be further speed up by implementing in parallel, e.g. using GPU, and possible to satisfy the requirements of real-time applications.

Fig. 11. Reconstructed frames with different methods. The test video sequence, *BQSquare*, is compressed by HEVC (AI) at QP=37. (a) HEVC-N. (b) HEVC-I. (c) SKR. (d) NLM. (e) SKR. (f) SA-DCT. (g) BM3D. (h) FoE. (i) CSR. (j) LPG-PCA. (k) Zhang's. (l) Proposed.

The proposed method takes advantage of temporal information, which will cause a delay of p frames (in our experiments $p = 2$) for real-time applications. Since most videos are of frame rate higher than 30fps, the delay incurred by our method is smaller than 0.06s, which is acceptable for most video applications.

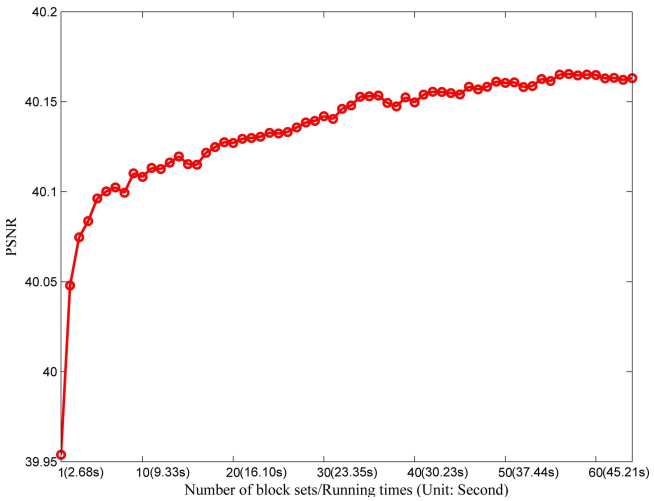


Fig. 12. PSNR performance vs. number of processed block sets and running time.

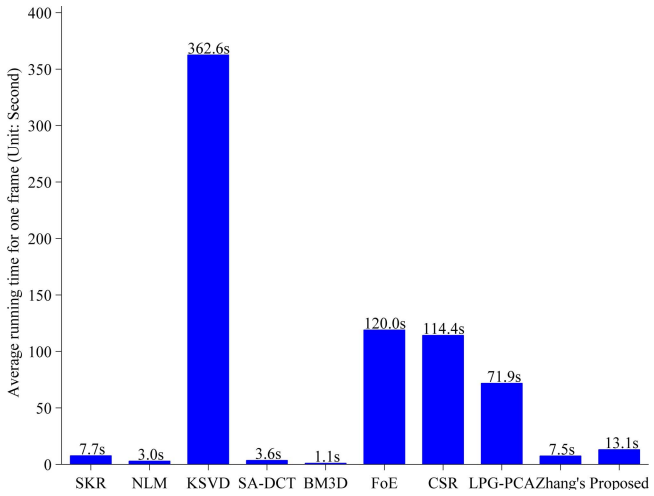


Fig. 13. Average running time of different methods for video sequences with CIF and WQVGA formats.

VI. Conclusion

In this paper, we propose a video compression artifact reduction method by adaptively fusing multiple hypotheses based on their reliabilities. The temporal AR model and non-local similar blocks are utilized in DCT domain to estimate the original coefficients. Finally, overlapped estimations are weighted and aggregated adaptively to generate high quality videos. Experimental results demonstrate that our proposed method can remarkably improve both the subjective and the objective quality of the compressed video sequences. The proposed method can be plugged into many video application systems after decoder module. For example, it can be plugged into video players to directly improve the video quality and user experience, and it also may be used as a preprocessing tool in transcoding application, which may help saving bits for the transcoded video streaming due to noise reduction.

References

[1] W. Gao, Y. Tian, T. Huang, S. Ma, and X. Zhang, "The IEEE 1857 standard: Empowering smart video surveillance systems," *IEEE Intell. Syst.*, vol. 29, no. 5, pp. 30–39, Sep./Oct. 2014.

[2] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[3] C.-M. Fu *et al.*, "Sample adaptive offset in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1755–1764, Dec. 2012.

[4] X. Zhang, R. Xiong, S. Ma, and W. Gao, "Adaptive loop filter with temporal prediction," in *Proc. Picture Coding Symp. (PCS)*, May 2012, pp. 437–440.

[5] C.-Y. Tsai *et al.*, "Adaptive loop filtering for video coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 934–945, Dec. 2013.

[6] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[7] I. H. Reeve, III, and J. S. Lim, "Reduction of blocking effect in image coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. ICASSP*, vol. 8. Apr. 1983, pp. 1212–1215.

[8] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 5, pp. 1258–1268, Oct. 1986.

[9] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2005, pp. 60–65.

[10] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

[11] G. Zhai, W. Zhang, X. Yang, W. Lin, and Y. Xu, "Efficient image deblocking based on postfiltering in shifted windows," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 122–126, Jan. 2008.

[12] G. Zhai, W. Lin, J. Cai, X. Yang, and W. Zhang, "Efficient quadtree based block-shift filtering for deblocking and deringing," *J. Vis. Commun. Image Represent.*, vol. 20, no. 8, pp. 595–607, Nov. 2009.

[13] S. Wu, H. Yan, and Z. Tan, "An efficient wavelet-based deblocking algorithm for highly compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 11, pp. 1193–1198, Nov. 2001.

[14] G. Zhai, W. Zhang, X. Yang, W. Lin, and Y. Xu, "Efficient deblocking with coefficient regularization, shape-adaptive filtering, and quantization constraint," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 735–745, Aug. 2008.

[15] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, May 2007.

[16] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[18] X. Zhang, R. Xiong, S. Ma, and W. Gao, "Reducing blocking artifacts in compressed images via transform-domain non-local coefficients estimation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2012, pp. 836–841.

[19] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4613–4626, Dec. 2013.

[20] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, no. 4, pp. 1531–1549, Apr. 2010.

[21] W. Dong, X. Li, D. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 457–464.

[22] J. Zhang, R. Xiong, C. Zhao, S. Ma, and D. Zhao, "Exploiting image local and nonlocal consistency for mixed Gaussian-impulse noise removal," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2012, pp. 592–597.

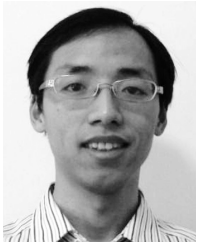
[23] J. Zhang, D. Zhao, R. Xiong, S. Ma, and W. Gao, "Image restoration using joint statistical modeling in a space-transform domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 915–928, Jun. 2014.

[24] X. Zhang, R. Xiong, S. Ma, and W. Gao, "Artifact reduction of compressed video via three-dimensional adaptive estimation of transform coefficients," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 4567–4571.

[25] 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, Dec. 2012.
[26] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[27] X. Zhang and X. Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 887–896, Jun. 2008.
[28] X. Zhang, S. Ma, Y. Zhang, L. Zhang, and W. Gao, "Nonlocal edge-directed interpolation," in *Proc. Conf. PCM*, vol. 5879. Jan. 2009, pp. 1197–1207.
[29] Y. Zhang, D. Zhao, X. Ji, R. Wang, and W. Gao, "A spatio-temporal auto regressive model for frame rate upconversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1289–1301, Sep. 2009.
[30] A. Norkin *et al.*, "HEVC deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012.
[31] D. Sun and W.-K. Cham, "Postprocessing of low bit-rate block DCT coded images based on a fields of experts prior," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2743–2751, Nov. 2007.
[32] *The Code HM 12.0*. [Online]. Available: https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags, accessed Jun. 2014.
[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

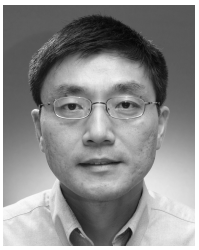
Xinfeng Zhang received the B.S. degree in computer science from the Hebei University of Technology, Tianjin, China, in 2007, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2014.

He is currently a Research Fellow with Nanyang Technological University, Singapore. His current research interests include image and video processing, and image and video compression.



Ruiqin Xiong (M'08) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2001, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2007.

He was a Research Intern with Microsoft Research Asia from 2002 to 2007 and a Senior Research Associate with the University of New South Wales, Australia, from 2007 to 2009. He joined Peking University, Beijing, in 2010. His research interests include statistical image modeling, image and video processing, video compression, and multimedia communication.



Weisi Lin (M'92–SM'98) received the B.Sc. degree in electronics and the M.Sc. degree in digital signal processing from Zhongshan University, Guangzhou, China, in 1982 and 1985, respectively, and the Ph.D. degree in computer vision from Kings College, London University, London, U.K., in 1992.

He was involved in teaching and research with Zhongshan University, Shantou University, Shantou, China, Bath University, Bath, U.K., the National University of Singapore, Institute of Microelectronics, Singapore, and the Institute for Infocomm Research, Singapore. He was the Laboratory Head of Visual Processing and the Acting Department Manager of Media Processing with the Institute for Infocomm Research. He is currently an Associate Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include image processing, perceptual modeling, video compression, multimedia communication, and computer vision.

Siwei Ma (M'12) 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.

He was a Post-Doctorate with the University of Southern California from 2005 to 2007. Then, he joined the School for Electrical Engineering and Computer Science, Institute of Digital Media, Peking University, where he is currently an Associate Professor. He has authored over 100 technical articles in refereed journals and proceedings in the areas of image and video coding, video processing, video streaming, and transmission.

Jiaying Liu (S'09–M'10) received the B.E. degree in computer science from Northwestern Polytechnic University, Xi'an, China, in 2005, and the Ph.D. (Hons.) degree in computer science from Peking University, Beijing, China, in 2010.

She was a Visiting Scholar with the University of Southern California, Los Angeles, from 2007 to 2008. She is currently an Associate Professor with the Institute of Computer Science and Technology, Peking University. Her current research interests include image processing, sparse signal representation, and video compression.

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