Low-Rank-Based Nonlocal Adaptive Loop Filter for High-Efficiency Video Compression

Xinfeng Zhang, *Member, IEEE*, Ruiqin Xiong, *Member, IEEE*, Weisi Lin, *Fellow, IEEE*, Jian Zhang, *Member, IEEE*, Shiqi Wang, *Member, IEEE*, Siwei Ma, *Member, IEEE*, and Wen Gao, *Fellow, IEEE*

***Abstract*— In video coding, the in-loop filtering has emerged as a key module due to its significant improvement on compression performance since H.264/Advanced Video Coding. Existing incorporated in-loop filters in video coding standards mainly take advantage of the local smoothness prior model used for images. In this paper, we propose a novel adaptive loop filter utilizing image nonlocal prior knowledge by imposing the low-rank constraint on similar image patches for compression noise reduction. In the filtering process, the reconstructed frame is first divided into image patch groups according to image patch similarity. The proposed in-loop filtering is formulated as an optimization problem with low-rank constraint for every group of image patches independently. It can be efficiently solved by soft-thresholding singular values of the matrix composed of image patches in the same group. To adapt the properties of the input sequences and bit budget, an adaptive threshold derivation model is established for every group of image patches according to the characteristics of compressed image patches, quantization parameters, and coding modes. Moreover, frame-level and largest coding unit-level control flags are signaled to further improve the adaptability from the sense of rate-distortion optimization. The performance of the proposed in-loop filter is analyzed when it collaborates with the existing in-loop filters in High Efficiency Video Coding. Extensive experimental results show that our proposed in-loop filter can further improve the performance of state-of-the-art video coding standard significantly, with up to 16% bit-rate savings.**

***Index Terms*— High Efficiency Video Coding (HEVC), in-loop filter, nonlocal, singular value decomposition (SVD), video coding.**

I. Introduction

BLOCK-BASED transforms have been widely used in most popular video coding standards, e.g., H.264/Advanced Video Coding (AVC) [1] and High Efficiency Video Coding (HEVC) [2], because it can exploit the correlation among pixels and hence improve the performance of video communication systems [3]. Due to coarse quantization and motion compensation, many compression artifacts, such as blocking, ringing, and blurring, are introduced at medium and low bit rates. In-loop filters have become essential video coding tools, which can not only improve the quality of decoded frames directly by reducing the compression artifacts but also provide high-quality reference frames for succeeding pictures to directly save coding bits by motion compensation.

The first in-loop filter adopted into video coding standards is deblocking filter (DF) [4] in H.264/AVC, which applies a set of low-pass filters to 4×4 block boundaries adaptively based on the characteristics of reconstructed samples on both sides of block boundaries, and coding information, such as prediction modes and motion vectors. Although additional computational complexity is introduced, DF can obviously improve the subjective quality of decoded videos and achieve significant bit-rate saving. The latest generation video coding standard, HEVC, also adopts the DF as the first in-loop filter, which is extended to be performed on 8×8 block boundaries to reduce the number of deblocking operations. Besides the DF specially designed for blocking artifacts, HEVC further adopts another nonlinear in-loop filter, sample adaptive offset (SAO) [5], to further reduce the coding artifacts after DF. SAO derives offset values for different regions of reconstructed frames from DF by minimizing mean sample distortions with original images, and modifies reconstructed samples by conditionally adding the corresponding offset value to each sample after the application of the DF. These offset values need to be coded and transmitted to decoders.

During the development of HEVC, different kinds of in-loop filters have been discussed. Adaptive loop filter (ALF) [6] had ever been adopted as the third in-loop filter after SAO in the early stage of HEVC, which is a Wiener-based adaptive filter to minimize the mean square errors between original samples and reconstructed samples from SAO. To adapt to different image structures, ALF derives up to 16 filters for different areas of luminance component. Similar to SAO, the derived filter parameters need to be transmitted to decoders, which will introduce a relatively large overhead. To achieve better coding performance, these areas are further merged based on rate-distortion (RD) costs at encoder side to reduce the number of filters, which can not only reduce the overhead of ALF but also decrease the adaptability of ALF for different image

Manuscript received October 11, 2015; revised January 14, 2016 and April 16, 2016; accepted June 11, 2016. Date of publication June 15, 2016; date of current version October 3, 2017. This work was supported in part by the Rapid-Rich Object Search Laboratory, Nanyang Technological University, Singapore, and Peking University, through the National Research Foundation, Singapore, under its Interactive Digital Media Strategic Research Programme, and in part by the National Basic Research Program of China (973 Program) under Grant 2015CB351800. This paper was recommended by Associate Editor M. Rabbani. *(Corresponding author: Weisi Lin.)*

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

R. Xiong, J. Zhang, 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; jian.zhang@pku.edu.cn; swma@pku.edu.cn; wgao@pku.edu.cn).

S. Wang is with the Department of Computer Science, City University of Hong Kong, Hong Kong. (e-mail: shiqwang@cityu.edu.hk).

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

Digital Object Identifier 10.1109/TCSVT.2016.2581618

contents. To ensure the improvement for coding efficiency, different ON/OFF control flags are designed in syntax, such as picture-level control, quadtree-based control [7], and largest coding unit (LCU)-level control [8], and the values that are determined based on RD costs at the encoder side. For example, the flag of picture-level control is true only if the RD costs decrease when ALF is applied to the current picture. Although ALF is well designed to adapt to different image contents, the adaptability of ALF is actually limited by the large overhead of side information (e.g., filter parameters and control flags).

To improve the performance of in-loop filters, substantial research has been proposed in studies [9]–[15]. Our previous work [9] takes advantage of temporal correlation between adjacent video frames to reduce the overhead of ALF by reusing the ALF parameters in previously encoded frames. In [10], a high-level syntax structure, adaptation parameter set (APS), is specially designed for picture-adaptive data, including ALF and SAO parameters, which makes in-loop filter parameters reuse more flexible with APS indices. Considering the high efficiency of well-known nonlocal means (NLM) filter [11] in denoising problem, Matsumura *et al.* [12] first introduced NLM filter into HEVC to compensate for the shortcomings of the existing in-loop filters based only on image local smoothness prior models. The NLM filter takes weighted average of nonlocal similar image patches to reduce compression artifacts in which the weights are determined by the similarity of image patches located at the source and target coordinates. The nonlocal-based in-loop filter is further improved by delicately designing patch shapes, search window shapes, and optimizing filter ON/OFF control module based on RD costs [13]. Han *et al.* [14] also utilized the nonlocal similar image patches in a quadtree-based Kuan's filter to suppress compression artifacts by adaptively combining the pixels restored by NLM filter and reconstructed pixels according to the variance of image signals and quantization noise. Many other denoising methods [16]–[23] based on nonlocal similarity are proposed as a postprocessing filter to improve image visual quality.

During recent years, image low-rank prior models have been widely used in image restoration field and have achieved significant improvement [24]–[27]. Guo *et al.* [24] utilized low-rank approximation to remove Gaussian noise with independent identical distribution. Ren *et al.* [25] explored image low-rank prior model and designed a postprocessing filter to reduce compression noise in JPEG images. Dong *et al.* [26] utilized low-rank constraint as a regularization term in compressive sensing to restore high-quality images. However, these methods usually use a global parameter, i.e., a threshold or a regularization parameter, to control the importance of low-rank constraint in objective functions, and the parameters need to be manually adjusted to get satisfactory results.

In this paper, we take advantage of image low-rank prior model and propose an ALF by collaboratively filtering nonlocal image patches with content-dependent parameters to improve the performance of video coding. The proposed in-loop filter is named nonlocal ALF (NALF). In the proposed method, a reconstructed picture is first divided into image patches, and for each patch, we find the K most similar patches to form an image patch group, which is rearranged into a matrix. The compression artifacts are reduced by jointly minimizing the mean square errors between reconstructed samples and latent original samples and the rank of a similar image patch matrix. The optimization problem is solved by soft thresholding the singular values of image patch matrix. We further derive adaptive parameters for every group of image patches according to the statistical characteristics of reconstructed samples and coding information, i.e., intra coding/inter coding and quantization parameters (QPs). To achieve better coding performance, we also add ON/OFF control flags in syntax at different levels, i.e., frame level and LCU level, which are determined based on RD costs of NALF at the encoder side. Finally, we integrate the proposed method into HEVC reference software at different positions relative to other in-loop filters, and analyze the impacts of the proposed in-loop filter on coding performance when it cooperates with others.

The remainder of this paper is organized as follows. Section II introduces the framework of the proposed low-rank-based NALF. Section III introduces the adaptive parameter derivation and ON/OFF control design for NALF. Extensive experimental results and analyses are reported in Section IV, and Section V concludes this paper.

II. Framework of Low-Rank-Based Nonlocal Adaptive Loop Filter

In this section, we first introduce the problem formulation of low-rank-based image denoising, and then introduce the framework of the proposed NALF in hybrid video coding.

A. Problem Formulation of Low-Rank-Based Image Denoising

In low-rank-based image denoising, a noisy image, $\mathcal{I}_{\boldsymbol{y}}$, is first divided into patches with the size of $p \times p$, which are denoted by center patches. These image patches are extracted every s pixels in raster scanning order, and s is referred to as neighbor distance, which is used to describe the extent of overlapping. For each noisy center patch, $\boldsymbol{y}_i$, we find K-nearest neighbors (KNNs) surrounding it according to the $L2$ norm distance of image patches

$$d(\boldsymbol{y}_i, \boldsymbol{y}_j) = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2. \tag{1}$$

These KNNs of $\boldsymbol{y}_i$ construct an image patch group, G_i, and are arranged into a matrix

$$\mathbf{Y}_{G_i} = \left[\boldsymbol{y}_{G_i}^{(1)}, \boldsymbol{y}_{G_i}^{(2)}, \ldots, \boldsymbol{y}_{G_i}^{(K)}\right] \tag{2}$$

where $\mathbf{Y}_{G_i}$ is a matrix with the size of $p^2 \times K$ by arranging every image patch in group G_i as a column vector. The image patch grouping procedure is shown in Fig. 1. The noise-free version of $\mathbf{Y}_{G_i}$ is denoted by $\mathbf{X}_{G_i}$

$$\mathbf{Y}_{G_i} = \mathbf{X}_{G_i} + \mathbf{N}_{G_i} \tag{3}$$

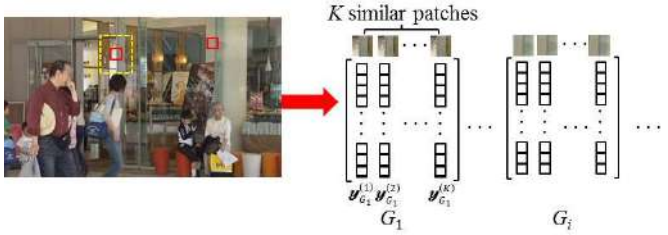


Fig. 1. Image patch grouping procedure, searching K similar patches for every center patch (red patch) in a neighborhood (yellow region) around it.

where $\mathbf{N}_{G_i}$ is the compression noise in these image patches. The denoising problem with low-rank constraint can be formulated for every group of image patches independently as

$$\min_{\mathbf{X}_{G_i}} \text{rank}(\mathbf{X}_{G_i}), \quad \text{s.t. } \|\mathbf{Y}_{G_i} - \mathbf{X}_{G_i}\|_2^2 \le \gamma\, \sigma_n \tag{4}$$

where $\text{rank}(\mathbf{X}_{G_i})$ represents the rank of matrix $\mathbf{X}_{G_i}$. σ_n is the standard deviation of noise and γ is a constant.

The rank of a matrix $\mathbf{X}$ is equal to the number of its nonzero singular values, i.e., the number of nonzero diagonal elements in matrix $\mathbf{\Lambda}$ of the singular value decomposition (SVD) of $\mathbf{X}$, as in

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^* \tag{5}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the two unitary square matrices ($\mathbf{V}^*$ is the conjugate transpose of $\mathbf{V}$). Therefore, $\text{rank}(\mathbf{X})$ is the $L0$ norm of the vector (named singular value vector) composed of diagonal elements of Λ, which makes the objective function in (4) an NP-hard problem. The nuclear norm of singular value vector of $\mathbf{X}_{G_i}$ is used instead of the $L0$ norm of that, which is a widely used strategy in solving sparse constraint problems [28]. The objective function in (4) is transformed as

$$\widehat{\mathbf{X}}_{G_i} = \arg\min_{\mathbf{X}_{G_i}} \|\mathbf{Y}_{G_i} - \mathbf{X}_{G_i}\|_2^2 + \gamma\, \|\mathbf{X}_{G_i}\|_* \tag{6}$$

where the second term is the nuclear norm [29] of the matrix $\mathbf{X}_{G_i}$, which is defined as

$$\|\mathbf{X}_{G_i}\|_* = \text{tr}\left(\sqrt{\mathbf{X}_{G_i}^* \mathbf{X}_{G_i}}\right) = \sum_{k=0}^{N} \lambda_k \tag{7}$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, λ_k is the kth diagonal element in the singular matrix Λ_{G_i} of $\mathbf{X}_{G_i}$, and N is the number of diagonal elements in Λ_{G_i}.

Many algorithms have been proposed to solve the optimization problem with low-rank constraint in (6) in the studies [28], [30], [31]. In this paper, we take the soft-thresholding method in [28] due to its computation efficiency. The problem in (6) is solved as follows:

$$\text{SVD}(\mathbf{Y}_{G_i}) = \mathbf{U}_{G_i} \Lambda_{G_i} \mathbf{V}_{G_i}^* \tag{8}$$

$$\Lambda_{G_i,\boldsymbol{\tau}} 0 = \text{SoftThr}(\Lambda_{G_i}, \boldsymbol{\tau}) \tag{9}$$

$$\widehat{\mathbf{X}}_{G_i} = \mathbf{U}_{G_i} \Lambda_{G_i,\boldsymbol{\tau}} \mathbf{V}_{G_i}^*. \tag{10}$$

To solve the optimization problem in (6), SVD is first applied to the image patch matrix, $\mathbf{Y}_{G_i}$, to generate matrices, $\mathbf{U}_{G_i}$, Λ_{G_i}, and $\mathbf{V}_{G_i}$ as in (8). Second, the soft-thresholding operation, $\text{SoftThr}(\cdot)$, is further applied to the singular value matrix, Λ_{G_i}, to derive the shrunken matrix, $\Lambda_{G_i,\boldsymbol{\tau}}$. Herein, $\text{SoftThr}(\Lambda_{G_i}, \boldsymbol{\tau})$ is a function that shrinks the diagonal elements of Λ_{G_i} according to the threshold $\boldsymbol{\tau}$. Specially, the kth diagonal element in Λ_{G_i} is shrunken by a nonlinear function, $D_{\boldsymbol{\tau}(k)}(\cdot)$, at level $\boldsymbol{\tau}(k)$

$$D_{\boldsymbol{\tau}_k}(\lambda_k) : \lambda_{k,\boldsymbol{\tau}(k)} = \max(|\lambda_k| - \boldsymbol{\tau}(k), 0). \tag{11}$$

The matrix $\Lambda_{G_i,\boldsymbol{\tau}(k)}$ is composed of the shrunken singular values, $\lambda_{k,\boldsymbol{\tau}(k)}$, at diagonal positions. Finally, the image patch matrix with noise mitigation is reconstructed via (10) with the shrunken singular value matrix, $\Lambda_{G_i,\boldsymbol{\tau}(k)}$. This procedure can be performed iteratively until a specified termination criterion is satisfied. Considering the computational complexity, we apply the above operations in (8)–(10) to every group only once, without iterations.

Since the image patches in groups are overlapped, there may be multiple filtered values for one sample. Therefore, the weighted average of multiple filtered sample values is utilized to reconstruct a high-quality image [25]

$$\widehat{\boldsymbol{x}}(i,j) = \sum_{k \in \Omega_{i,j}} w_k \widehat{\boldsymbol{x}}_k(i',j') \tag{12}$$

where w_k is the weight for filtered samples in image patch $\widehat{\boldsymbol{x}}_k$, and $\Omega_{i,j}$ is a set composed of the image patches that include the image pixel $\widehat{\boldsymbol{x}}(i,j)$. Coordinates (i,j) and (i',j') indicate the same sample under the whole image and the image patch coordinate systems, respectively. The weight, w_k, is determined according to the rank of the matrix composed of similar image patches in one group

$$w_k = \frac{1}{S} \max\left(\left(1 - \frac{r}{W_{G_i}}\right), \frac{1}{W_{G_i}}\right), \quad \boldsymbol{x}_k \in G_i \tag{13}$$

where r is the rank of the matrix $\mathbf{Y}_{G_i}$, and S is a normalized constant. Here, W_{G_i} is the minimum dimension of matrix $\mathbf{Y}_{G_i}$

$$W_{G_i} = \min(p^2, K). \tag{14}$$

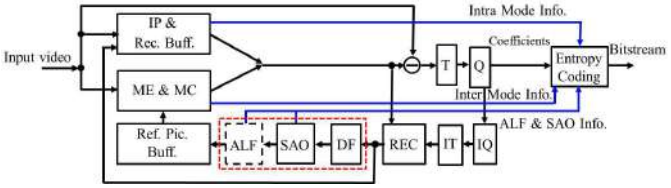
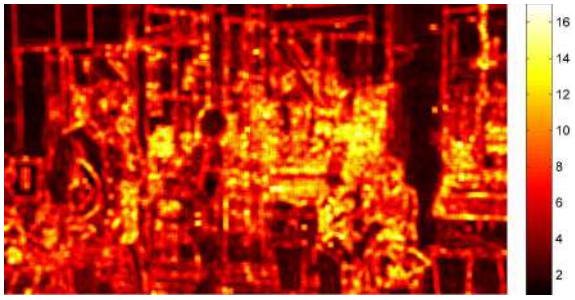


Fig. 2. Framework of HEVC (ALF has been removed from HEVC standard).

B. Framework of the Proposed Nonlocal Adaptive Loop Filter

Fig. 2 shows the framework of HEVC, where ALF has been removed from the final HEVC standard. There are three loop filters, i.e., DF, SAO, and ALF, which take effects sequentially on reconstructed pictures. Our proposed NALF is also a sequential one between reconstructed picture module and reference picture buffer module. Specially, the proposed NALF can achieve better performance when placed between DF and SAO based on our experimental results. We denote the

(a) decoded image, *BQMall*, size of 832x480



(b) standard deviation image of compression noise

Fig. 3. Standard deviations of compression noise in one video frame of *BQMall* compressed by HEVC at QP = 37. (a) Decoded video frame. (b) Standard deviation image of compression noise.

pixels in the input image of our filter as reconstructed samples, and the output as filtered samples in the following discussion.

Chang *et al.* [30] have derived the optimal thresholds for signals with different distributions

$$\tau = \frac{c\sigma_n^2}{\sigma_x} \tag{15}$$

where σ_n and σ_x are the standard deviation of noise and original signals, respectively. The parameter, c, is a constant and varies with original signal distributions, e.g., $c = 1$ for Gaussian distribution signals and $c = 2\sqrt{2}$ for Laplace distribution. Although image signals cannot be accurately formulated by only one distribution ideally, the optimal threshold in (15) also points out that the reasonable thresholds are related to the standard deviations of noise and signals.

Since existing image denoising work with low-rank constraint mainly focuses on dealing with general noise, such as Gaussian noise and impulse noise, the standard deviation of noise is simply utilized as a global parameter to control the filtering strength, which is assumed to be known or estimated based on the assumption that the noise is independent of image signals and follows identical distribution in the entire image [32]–[34]. However, compression noise is obviously correlated with image signals, which is mainly caused by quantizing transform coefficients in block-based hybrid video coding. Fig. 3 shows the different standard deviations of compression noise in a video frame compressed by HEVC intra coding at QP = 37. The standard deviation of compression noise for every decoded pixel is calculated in a 6×6 neighborhood centered at the corresponding decoded pixel. We can see that the compression noise level varies significantly with different image contents, and higher

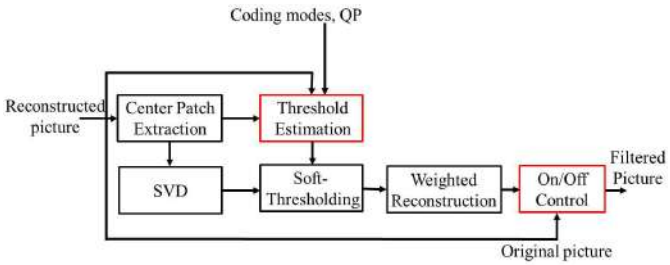


Fig. 4. Framework of the proposed NALF, with the red boxes highlighting our contributions.

noise levels usually distribute around image edges and texture regions, while lower noise levels usually exist in smooth areas. In addition, another obvious phenomenon is that there are approximate noise levels in image area with similar structures, e.g., along edges.

Therefore, we propose an NALF based on the low-rank constraint, which estimates threshold for every group of image patches with the following formulation:

$$\boldsymbol{\tau}_{G_i}(k) = \frac{c\sigma_{n,G_i}^2}{\sigma_{x,G_i,k}} \tag{16}$$

where σ_{n,G_i} is the standard deviation of compression noise, and $\sigma_{x,G_i,k}$ is the standard deviation of original image patches in the kth dimension of SVD space for group G_i, respectively. In the proposed method, we estimate the standard deviation of compression noise based on three factors, i.e., the statistical characteristics of reconstructed samples, QPs, and coding modes (i.e., intra coding/inter coding), and find the best global parameter c for different coding modes. Since the proposed filter cannot ensure the reduction of compression noise for every image patch, different ON/OFF control flags are utilized to switch the filtering process based on RD costs. The diagram of the proposed NALF is shown in Fig. 4, and the details of the modules in the red boxes are introduced in Section III.

III. Content-Dependent Threshold Derivation and Syntax Design for Filtering Control

A. Content-Dependent Threshold Derivation

Based on the discussion in Section II-B, the standard deviation of compression noise is obviously related to image content and QP, because the coefficient quantization is one of the main sources for compression noise. In this paper, we utilize the average standard deviation of all the image patches in one group to describe the characteristics of image content, while the standard deviation of compression noise is utilized to reflect compression noise levels. A statistical relationship of compression noise level, statistics of image content, and QPs is shown in Fig. 5 in which the video sequences are compressed by HEVC intra coding. We can see that the compression noise level (described by standard deviation of compression noise) is proportional to the standard deviation of image patches and QP. The compression noise level is higher with larger QPs. With the same QP, the compression noise level increases with the standard deviation of image patches. This is because the image patches with smaller standard deviations are usually in

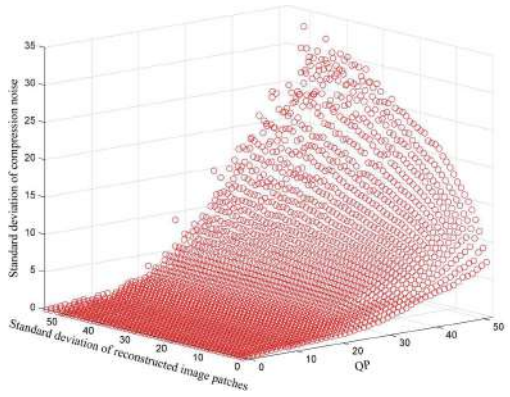


Fig. 5. Relationship of compression noise levels, statistics of image content, and QPs for video sequences compressed by HEVC intra coding.

smooth areas, whereas most of the prediction residuals are very small. However, the 3D relationship is difficult to formulate accurately with a simple function. Fortunately, there are finite QPs for video coding standards. Therefore, we formulate the relationship of the compression noise level and the standard deviation of image content for every QP.

Besides quantization, inter prediction with motion compensation is also one of the sources introducing compression noise, which may directly copy the previous decoded blocks for reconstruction, e.g., skip mode. Since these copied blocks may have been filtered by loop filters, the compression noise in inter-coding frames may be different from that in intra-coding frames. Therefore, we explore the compression noise levels in different frame types (i.e., I/P/B frame). For I frame, there is only intra prediction, and the compression noise is generated only by quantization, whereas for P/B frame, besides intra prediction, there are unidirectional and bidirectional inter predictions, which may introduce compression noise from reference frames.

We learn the relationship between compression noise level and standard deviation of similar image patches from a few common test sequences. These sequences are compressed by HEVC under three widely used coding configures, i.e., all intra (AI) coding, low delay (LD) coding, and random access (RA) coding. In AI, there is only intra prediction for all the frames. In LD, the unidirectional inter prediction (P frame) from the previous coded frames is utilized, while in RA, the hierarchical bidirectional inter prediction (B frame) is utilized. The standard deviation of a group of similar image patches is estimated by averaging all the reconstructed image patches in one group as follows:

$$\sigma_y = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\sum_{i=1}^{p^2}\frac{(\boldsymbol{y}_k(i)-\mu_k)^2}{p^2}} \tag{17}$$

$$\mu_k = \frac{1}{p^2}\sum_{i=1}^{p^2}\boldsymbol{y}_k(i) \tag{18}$$

where $\boldsymbol{y}_k(i)$ represents the ith pixel of image patch vector $\boldsymbol{y}_k$.

Fig. 6 shows the fitting results for compression noise level and standard deviation of reconstructed image patches by HEVC at QP = 32 and 37 under AI, LD, and RA coding configurations, respectively. We can see that the power function ($y = ax^b$) can well fit the relationship of compression noise levels (y) and standard deviations of reconstructed image patches (x) at a constant QP. By comparing the fitting results at the same QP, we can see that the compression noise level in compressed videos with inter prediction (LD and RA) is usually smaller than that of images compressed with intra prediction (AI) because the prediction efficiency of inter prediction is much higher than that of intra prediction, and the prediction residuals are more centralized to zeros, which makes smaller compression noise level at the same QP. We can further estimate the compression noise level with the learned parameters for all the QPs from 0 to 51 in HEVC and three frame types (I/P/B frame). Tables I and II show the parts of the fitting results for luminance and chroma components at typical QPs utilized in HEVC common test, respectively.

TABLE I

PARAMETERS OF NOISE-LEVEL PREDICTION FUNCTION FOR LUMINANCE COMPONENT WITH DIFFERENT FRAME TYPES

QP	22		27		32		37	
Parameters	a	b	a	b	a	b	a	b
I frame	1.46	0.12	1.91	0.19	2.36	0.29	3.14	0.34
P frame	1.44	0.17	1.72	0.29	1.90	0.42	2.52	0.46
B frame	1.27	0.22	1.41	0.35	1.64	0.45	2.26	0.49

TABLE II

PARAMETERS OF NOISE-LEVEL PREDICTION FUNCTION FOR CHROMA COMPONENT WITH DIFFERENT FRAME TYPES

QP	22		27		32		37	
Parameters	a	b	a	b	a	b	a	b
I frame	1.21	0.20	1.74	0.24	2.24	0.28	3.07	0.23
P frame	1.16	0.26	1.59	0.31	2.02	0.33	2.63	0.30
B frame	1.08	0.29	1.58	0.29	2.24	0.26	2.57	0.30

The global parameter, c, in (16) also plays an important role in the performance of compression noise reduction. Since the inter coding takes the previous reconstructed frames as references, which have been processed by a series in-loop filters (DF, SAO, and ALF), the thresholds for inter-coding frames should be smaller than that in intra-coding frames to avoid oversmoothing. We also directly learn the best global parameter for different coding methods (i.e., intra coding and inter coding) based on the mean square error reduction of filtered images. Based on the statistical results, we find that it is useful for most cases when setting $c = 6$ and 8 for luminance component and chroma component with intra-coding frame, and $c = 0.1$ and 0.05 for luminance component and chroma component with inter-coding (LD and RA) frame.

In addition, the standard deviation of original image patches in SVD space is estimated as

$$\sigma_{\boldsymbol{x},G_i,k} = \sqrt{\max\left(\frac{\lambda_{G_i,k}^2}{W_{G_i}} - \sigma_{n,G_i}^2, 0\right)} \tag{19}$$

where $\lambda_{G_i,k}$ is the kth singular value of $\mathbf{Y}_{G_i}$. When $\sigma_{\boldsymbol{x},G_i,k}$ is zero, we skip the soft-thresholding operation for the corresponding singular values.

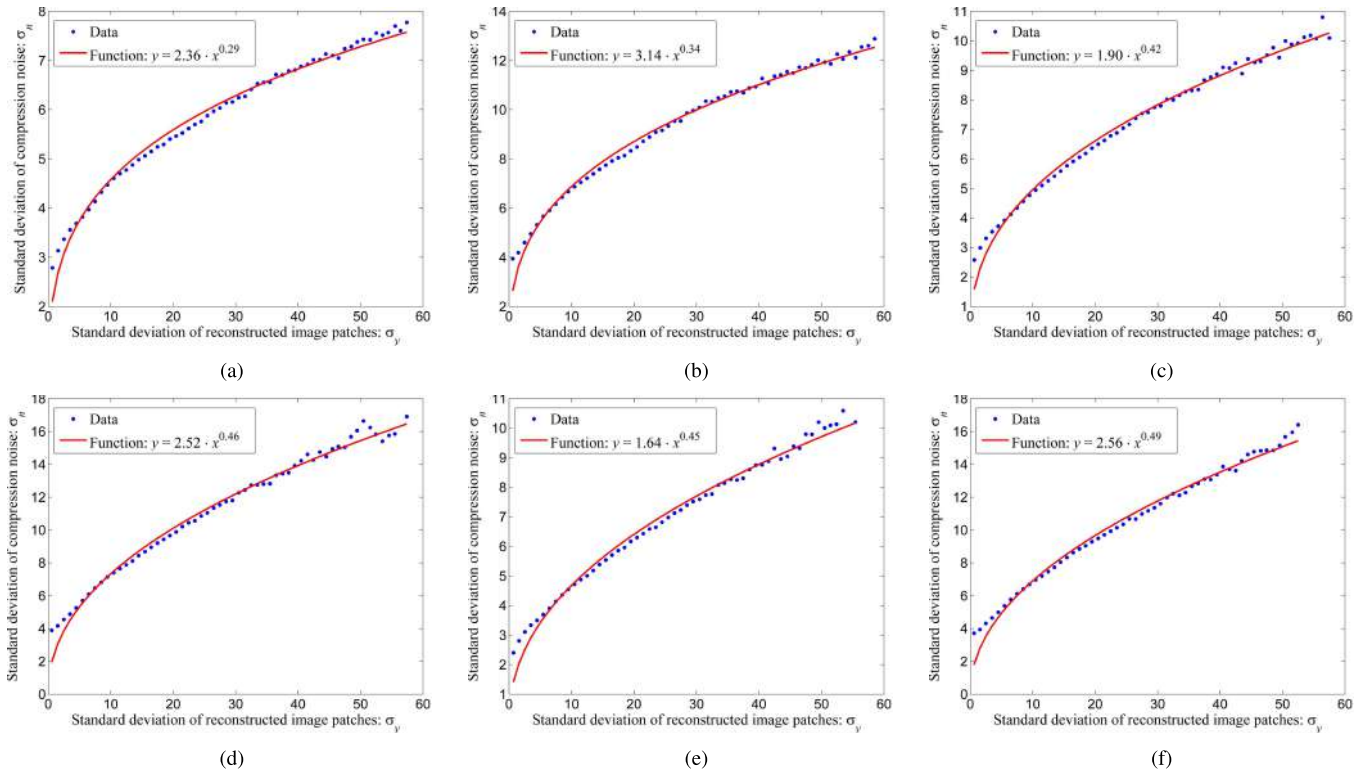


Fig. 6. Relationship of compression noise level and standard deviation of compressed image patches with HEVC at QP = 32 and 37 under AI, LD, and RA configurations. (a) HEVC AI coding, QP = 32. (b) HEVC AI coding, QP = 37. (c) HEVC LD coding, QP = 32. (d) HEVC LD coding, QP = 37. (e) HEVC RA coding, QP = 32. (f) HEVC RA coding, QP = 37.

B. Syntax Design for Filtering ON/OFF Control

Although the proposed NALF can reduce compression noise, it is difficult to ensure distortion reduction for every image patch. Therefore, to get better overall performance for video coding, we design ON/OFF control flags with two levels, i.e., LCU-level and frame-level control flags, for the proposed NALF. For LCU-level control, we add a flag for each LCU to switch NALF, i.e., **LCU_NALF_on**[i] for the ith LCU. When the distortions of filtered LCU decrease, the corresponding control flag is true, indicating that the NALF is applied to this LCU, while the distortions of filtered LCU increase, its control flag is false, indicating that the NALF is not applied to this LCU. After the determination of all the LCUs in one image, the reduction of RD costs is calculated as

$$J_c = \sum_{i=0}^{N} \delta(i)\left(d'_{c,i} - d_{c,i}\right) + \lambda_c R_c \tag{20}$$

where J_c is the RD cost reduction of the cth color component, and N is the amount of LCUs. $\delta(i)$ is an indicator function, which is equal to **LCU_NALF_on**[i]. $d_{c,i}$ and $d'_{c,i}$ are the mean square errors of the ith reconstructed LCU and filtered LCU. R_c is the amount of bits used to represent these LCU control flags, and λ_c is the Lagrange factor. If J_c is smaller than zero, a frame-level flag (**Y_NALF_on**, **U_NALF_on**, or **V_NALF_on** corresponding to luminance and chroma components) is true indicating that the NALF is applied to current frame. Furthermore, the corresponding frame-level and LCU-level control flags are encoded. If J_c is larger than zero, the frame-level flag is false, indicating that the NALF is not applied to current frame, and the LCU-level control flags will not be encoded further.

TABLE III
SYNTAX STRUCTURE FOR THE NALF CONTROL FLAGS

NALF_parameter_set() {	**Descriptor**
Y_NALF_on	1 bit
U_NALF_on	1 bit
V_NALF_on	1 bit
if(**Y_NALF_on**)	
for(i=0; i <*NumofLCU*; i++)	
LCU_NALF_on[i]	1 bit
}	

In addition to LCU-level control, we also explore NALF only with frame-level control flags. If the frame-level control flag is true, all the LCUs need to be filtered with NALF implicitly. Based on the compression performance, we propose to take LCU-level control for luminance component and frame-level control for chroma components in the final design of NALF. The syntax structure of the NALF control flags is shown in Table III.

Finally, we summarize the proposed algorithm at decoder side with pseudocode in Algorithm 1.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we investigate the performance of the proposed NALF by integrating it into HEVC reference software HM7.0, which has three in-loop filters,

TABLE IV
EXPERIMENTAL RESULTS OF NALF WITH FRAME-LEVEL CONTROL AT DIFFERENT POSITIONS UNDER AI CODING, ANCHOR IS HM7.0

Sequences	NALF-I			NALF-II			NALF-III			NALF-IV		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
Class B	-3.1%	-3.0%	-4.4%	-2.5%	-2.2%	-3.6%	-1.8%	-2.0%	-3.6%	-0.7%	-1.0%	-2.1%
Class C	-4.3%	-5.2%	-7.8%	-3.7%	-3.9%	-6.5%	-3.0%	-3.9%	-6.6%	-1.7%	-1.9%	-3.9%
Class D	-3.8%	-3.9%	-5.6%	-3.1%	-3.0%	-4.6%	-2.2%	-2.8%	-4.4%	-1.0%	-0.8%	-1.6%
Class E	-6.4%	-5.4%	-3.9%	-6.1%	-5.0%	-3.9%	-5.5%	-5.1%	-4.1%	-3.9%	-3.4%	-2.3%
Class F	-4.9%	-6.0%	-7.0%	-4.6%	-5.1%	-5.9%	-4.6%	-5.3%	-6.2%	-3.4%	-3.4%	-4.1%
Class G	-5.5%	-1.4%	-2.5%	-5.2%	-2.2%	-2.7%	-4.7%	-2.6%	-3.1%	-3.4%	-1.6%	-1.3%
Overall	-4.5%	-4.2%	-5.3%	-4.0%	-3.5%	-4.6%	-3.4%	-3.5%	-4.7%	-2.2%	-1.9%	-2.6%

Algorithm 1: Nonlocal Adaptive Loop Filter

Input: Reconstructed frame in previous coding stage, $\mathcal{I}_{\boldsymbol{y}}$.
while colorIdx *in (*Y/U/V*)* **do**
  Patch Extraction: Divide image into $p \times p$ *center patches* with *neighbor distance* s and the number of *center patches* is L;
  Patch Match: Find K-nearest neighbors for every *center patch* based on the similarity measure in Eqn. (1);
  for $i = 1$ *to* L **do**
    Noise Estimation: Calculate σ^2_{n,G_i} for image patch group, G_i, based on the fitting parameters and standard deviation of noisy patches in G_i;
    SVD: Apply SVD to image patch matrix $\mathbf{Y}_{G_i}$;
    Soft-Thresholding: Shrink the singular values of $\mathbf{Y}_{G_i}$ according to the thresholds derived by Eqns.(16) and (19);
    Patch Reconstruction: Reconstruct image patches with shrunken singular values as Eqn.(10);
  end
  Image Reconstruction: Weighted average all the estimated image patches via Eqn.(12);
  if colorIdx is Y **then**
    LCU Filter Control: Update reconstructed pixels with the filtered values according to LCU control flags;
  end
  Output: Decoded high quality image, $\mathcal{I}_{\hat{\boldsymbol{x}}}$
end

i.e., DF, SAO, and ALF. The widely used video sequences in HEVC standard development are utilized as test sequences in our experiments. There are 23 video sequences in our experiments, which are classified into six classes, Class B∼F is the same as that in the common test conditions (CTCs) in [35]. In addition, we add three conference video sequences with the resolution of 1280×720, which were also used in HEVC test and denoted by class G in this paper. For every sequence, the first 100 frames are encoded at four typical QPs, 22, 27, 32, and 37. Coding tools included in main conditions are enabled. Three video coding configurations are utilized in our experiments, i.e., AI, LD, and RA. The anchor is the performance of HM7.0 with all the three in-loop filters, and the other test conditions are the same with that in CTC. The coding efficiency is measured by using the Bjontegaard-Delta bit-rate measure [36]. The size of image patches is set to $p = 6$. For one frame, we extract the center patches every s ($s = 5$) pixels in raster scanning order. Therefore, the two neighboring center patches are overlapped with one column/row pixels. The center patch set of one frame can be formulated as

$$\Omega = \{\boldsymbol{x}_{m,n} | m, n \equiv 0 (\text{mod } s),\ 0 \le m < H - p \quad 0 \le n < W - p\} \quad (21)$$

where $\boldsymbol{x}_{m,n}$ is a $p \times p$ image patch, and H and W are the image height and width, respectively. For each center patch, the number of nearest neighbors is $K = 30$, which are searched exhaustively surrounding the current center patch with the search range $[-20\ 20]$ in both horizontal and vertical directions, respectively.

Since there have been three in-loop filters in HEVC, we try out the four positions, i.e., before DF, between DF and SAO, between SAO and ALF, and after ALF, to integrate our proposed NALF, which are denoted by NALF-I, NALF-II, NALF-III, and NALF-IV, respectively. Based on their performances, we can analyze the impacts of other in-loop filters on the proposed NALF and find the best position for our proposed NALF. Tables IV–VI show the overall performance of the proposed NALF with only frame-level control at different positions. No matter where it is placed, NALF can always improve the coding performance of HEVC significantly. For AI coding, NALF-I can achieve overall 4.5% bit-rate saving for the luminance component on all the sequences compared with that of HEVC, and especially, it achieves 6.4% bit-rate saving for class E on average. For LD and RA cases, NALF-II achieves the most bit-rate saving for the luminance component on all the test sequences, 5.3% and 4.7% on average, respectively. The proposed NALF achieves better results when integrated before DF and directly after DF than that of after SAO and after ALF because SAO and ALF may have smoothed out image textures to some extent while filtering out compression noise. Considering that inter coding is more frequently used than intra coding, we place the proposed NALF directly after DF to achieve better coding efficiency.

To analyze reasonable ON/OFF control mechanism, we further compare the coding performance of NALF-II with LCU-level control for only luminance component (Y) and all the three color components (YUV). Tables VII–IX show the coding gain of the proposed NALF-II with LCU-level control,

TABLE V

EXPERIMENTAL RESULTS OF NALF WITH FRAME-LEVEL CONTROL AT DIFFERENT POSITIONS UNDER LD CODING, ANCHOR IS HM7.0

Sequences	NALF-I			NALF-II			NALF-III			NALF-IV		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
Class B	-3.8%	-4.7%	-4.9%	-5.1%	-5.4%	-5.3%	-5.1%	-5.9%	-6.1%	-5.0%	-5.1%	-5.1%
Class C	-3.3%	-4.2%	-5.8%	-4.0%	-4.8%	-6.4%	-3.7%	-5.3%	-7.2%	-3.4%	-4.1%	-6.2%
Class D	-3.3%	-2.2%	-2.7%	-3.6%	-2.2%	-2.9%	-3.4%	-2.8%	-3.3%	-2.9%	-2.8%	-2.6%
Class E	-9.6%	-11.2%	-8.1%	-11.2%	-10.3%	-7.5%	-11.2%	-11.4%	-8.5%	-10.8%	-10.3%	-8.0%
Class F	-2.5%	-4.8%	-5.2%	-3.1%	-4.4%	-5.2%	-3.8%	-4.8%	-5.6%	-3.3%	-3.3%	-4.2%
Class G	-5.3%	-6.5%	-6.2%	-6.7%	-6.4%	-5.8%	-7.0%	-7.1%	-6.9%	-7.1%	-7.4%	-6.7%
Overall	-4.4%	-5.3%	-5.3%	-5.3%	-5.3%	-5.4%	-5.4%	-5.9%	-6.1%	-5.1%	-5.2%	-5.3%

TABLE VI

EXPERIMENTAL RESULTS OF NALF WITH FRAME-LEVEL CONTROL AT DIFFERENT POSITIONS UNDER RA CODING, ANCHOR IS HM7.0

Sequences	NALF-I			NALF-II			NALF-III			NALF-IV		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
Class B	-3.9%	-5.1%	-4.4%	-4.5%	-5.0%	-4.4%	-3.9%	-5.2%	-4.6%	-3.2%	-4.2%	-3.5%
Class C	-3.0%	-5.5%	-7.2%	-3.4%	-4.8%	-6.8%	-2.9%	-5.1%	-7.0%	-2.2%	-3.8%	-5.1%
Class D	-3.1%	-4.2%	-4.4%	-3.2%	-3.1%	-3.8%	-2.4%	-3.2%	-3.4%	-1.8%	-2.2%	-1.8%
Class E	-8.6%	-9.7%	-7.8%	-8.7%	-8.8%	-6.9%	-8.1%	-8.8%	-7.1%	-6.6%	-6.8%	-5.4%
Class F	-3.5%	-6.5%	-7.6%	-3.4%	-5.5%	-6.2%	-3.2%	-5.3%	-6.0%	-2.6%	-4.1%	-4.9%
Class G	-6.5%	-5.1%	-5.0%	-6.3%	-5.5%	-5.4%	-5.9%	-5.8%	-6.0%	-5.1%	-4.6%	-4.4%
Overall	-4.5%	-5.8%	-6.0%	-4.7%	-5.3%	-5.5%	-4.2%	-5.4%	-5.6%	-3.4%	-4.2%	-4.1%

TABLE VII

EXPERIMENTAL RESULTS OF NALF-II (LCU-Y) AND NALF-II (LCU-YUV) UNDER AI CODING, ANCHOR IS HM7.0

Sequences	NALF-II(LCU-Y)			NALF-II(LCU-YUV)		
	Y	U	V	Y	U	V
Class B	-2.5%	-2.2%	-3.5%	-2.3%	-2.4%	-3.7%
Class C	-3.7%	-3.9%	-6.4%	-3.6%	-4.0%	-6.4%
Class D	-3.1%	-3.0%	-4.5%	-3.0%	-2.8%	-4.4%
Class E	-6.0%	-4.9%	-3.8%	-5.6%	-4.9%	-3.6%
Class F	-4.5%	-5.0%	-5.8%	-4.3%	-4.9%	-5.6%
Class G	-5.2%	-2.1%	-2.7%	-4.8%	-1.9%	-2.7%
Overall	-4.0%	-3.5%	-4.5%	-3.8%	-3.4%	-4.5%

TABLE VIII

EXPERIMENTAL RESULTS OF NALF-II (LCU-Y) AND NALF-II (LCU-YUV) UNDER LD CODING, ANCHOR IS HM7.0

Sequences	NALF-II(LCU-Y)			NALF-II(LCU-YUV)		
	Y	U	V	Y	U	V
Class B	-6.4%	-4.4%	-4.6%	-5.9%	-7.3%	-7.6%
Class C	-5.6%	-4.4%	-5.8%	-5.1%	-8.1%	-10.5%
Class D	-4.4%	-1.7%	-2.2%	-4.2%	-4.7%	-4.7%
Class E	-11.6%	-8.9%	-5.5%	-10.7%	-13.1%	-10.2%
Class F	-5.1%	-3.4%	-4.0%	-4.6%	-7.4%	-7.8%
Class G	-8.9%	-5.6%	-4.5%	-8.2%	-7.2%	-7.4%
Overall	-6.7%	-4.5%	-4.4%	-6.2%	-7.8%	-7.9%

TABLE IX

EXPERIMENTAL RESULTS OF NALF-II (LCU-Y) AND NALF-II (LCU-YUV) UNDER RA CODING, ANCHOR IS HM7.0

Sequences	NALF-II(LCU-Y)			NALF-II(LCU-YUV)		
	Y	U	V	Y	U	V
Class B	-5.2%	-4.5%	-4.0%	-4.8%	-5.1%	-4.8%
Class C	-4.5%	-4.3%	-6.4%	-4.2%	-6.1%	-8.0%
Class D	-3.6%	-2.8%	-3.4%	-3.5%	-3.7%	-4.1%
Class E	-9.4%	-8.1%	-6.0%	-8.7%	-9.7%	-7.3%
Class F	-4.4%	-5.2%	-5.9%	-4.1%	-6.1%	-6.9%
Class G	-7.7%	-4.7%	-4.7%	-7.2%	-5.1%	-5.8%
Overall	-5.5%	-4.8%	-5.0%	-5.2%	-5.8%	-6.0%

where NALF-II (LCU-Y) and NALF-II (LCU-YUV) denote that the LCU control is applied only to luminance component and to all the three color components, respectively. First, by comparing LCU-level control (Tables VII–IX) and frame-level control (Tables IV–VI), we can see that the LCU-level control can further improve compression performance, obviously. NALF-II (LCU-Y) achieves about 1.4% (LD) and 0.8% (RA) bit-rate saving for luminance component on average compared with that of frame-level control. Since the overhead increases with LCU control for chroma components, NALF-II (LCU-YUV) achieves only 0.9% (LD) and 0.5% (RA) bit-rate saving for luminance component on average compared with that of frame-level control. Although there are higher bit-rate saving in NALF-II (LCU-YUV) for chroma components, we think it is better to apply LCU-level control for luminance component and frame-level control for chroma component due to the importance of luminance component.

Table X shows the detailed results of the proposed NALF with LCU control only in luminance component sequence by sequence when the ALF is OFF. Our proposed NALF achieves significant improvement on all these sequences; especially, for LD configure, the bit-rate savings are more than 10% for a few video sequences, e.g., *BQTerrace*, *FourPeople*, *Johnny*, *KristenAndSara*, *Vidyo1*, and *Vidyo3*. For *Johnny*, our method achieves more than 16% bit-rate saving for luminance component, because there are many repeated patterns in this sequence, e.g., the lines in coat, which are suitable for collaboratively filtering similar image patches. Fig. 7 shows the Peak signal-to-noise ratio (PSNR) curves for sequences, *Johnny*, and *KristenAndSara*, under LD and RA configures,

TABLE X

EXPERIMENTAL RESULTS OF THE PROPOSED NALF-II (LCU-Y) WITHOUT ALF IN SEQUENCE BY SEQUENCE, ANCHOR IS HM7.0 WITHOUT ALF

Sequences		AI			LD			RA		
		Y	U	V	Y	U	V	Y	U	V
Class B	Kimono	-4.2%	-0.9%	-0.9%	-11.3%	-11.2%	-7.3%	-4.8%	-5.4%	-2.4%
	ParkScene	-1.9%	0.2%	-0.3%	-4.1%	0.7%	-0.6%	-2.3%	-0.9%	0.6%
	Cactus	-4.0%	-2.2%	-7.0%	-9.7%	-3.7%	-3.0%	-7.2%	-5.4%	-5.1%
	BasketballDrive	-3.4%	-7.7%	-10.1%	-9.3%	-13.0%	-14.2%	-5.8%	-9.1%	-10.6%
	BQTerrace	-3.4%	-2.1%	-2.5%	-15.7%	-7.1%	-6.6%	-10.4%	-3.2%	-2.8%
Class C	BasketballDrill	-5.9%	-9.6%	-13.9%	-6.8%	-5.6%	-9.4%	-4.8%	-6.6%	-10.5%
	BQMall	-4.6%	-4.0%	-6.1%	-8.7%	-5.3%	-8.2%	-5.9%	-3.0%	-5.1%
	PartyScene	-1.5%	-0.2%	-0.8%	-2.2%	2.5%	3.1%	-1.5%	2.7%	3.3%
	RaceHorsesC	-2.3%	-2.8%	-7.8%	-6.3%	-7.4%	-11.4%	-4.7%	-5.8%	-11.1%
Class D	BasketballPass	-4.0%	-4.4%	-5.3%	-6.6%	-3.2%	-3.7%	-4.4%	-2.7%	-3.7%
	BQSquare	-1.8%	-1.2%	-3.8%	-2.1%	2.2%	2.1%	-0.8%	2.3%	0.3%
	BlowingBubbles	-1.9%	-3.8%	-4.6%	-3.1%	0.6%	1.3%	-2.3%	-0.4%	-0.3%
	RaceHorses	-3.5%	-5.3%	-7.5%	-6.1%	-4.2%	-4.5%	-4.9%	-5.4%	-5.8%
Class E	FourPeople	-6.8%	-6.9%	-7.3%	-11.4%	-9.5%	-8.4%	-9.9%	-9.2%	-8.6%
	Johnny	-7.8%	-12.5%	-10.9%	-16.8%	-20.1%	-15.3%	-10.7%	-15.9%	-13.4%
	KristenAndSara	-7.1%	-10.1%	-10.2%	-12.4%	-13.8%	-12.9%	-10.0%	-11.5%	-12.3%
Class F	BasketballDrillText	-5.6%	-8.6%	-11.4%	-6.4%	-3.4%	-5.2%	-4.7%	-6.5%	-8.4%
	ChinaSpeed	-2.2%	-2.1%	-1.8%	-3.2%	3.2%	3.2%	-3.1%	-0.8%	-1.0%
	SlideEditing	-4.1%	0.0%	-0.6%	-3.5%	-1.1%	-1.2%	-3.5%	0.0%	-0.3%
	SlideShow	-5.6%	-8.0%	-8.1%	-6.6%	-8.6%	-7.0%	-4.7%	-8.4%	-8.5%
Class G	vidyo1	-7.4%	-6.9%	-8.3%	-11.9%	-11.4%	-10.5%	-9.3%	-9.7%	-10.4%
	vidyo3	-6.8%	-2.8%	-3.2%	-15.6%	-11.9%	-5.6%	-10.0%	-6.8%	-3.6%
	vidyo4	-4.6%	-6.6%	-7.4%	-9.2%	-13.3%	-12.9%	-6.5%	-10.0%	-10.2%
Overall		-4.4%	-4.7%	-6.1%	-8.2%	-6.3%	-6.0%	-5.8%	-5.3%	-5.7%
Encoding Time		447%			206%			206%		
Decoding Time		17239%			20903%			13221%		

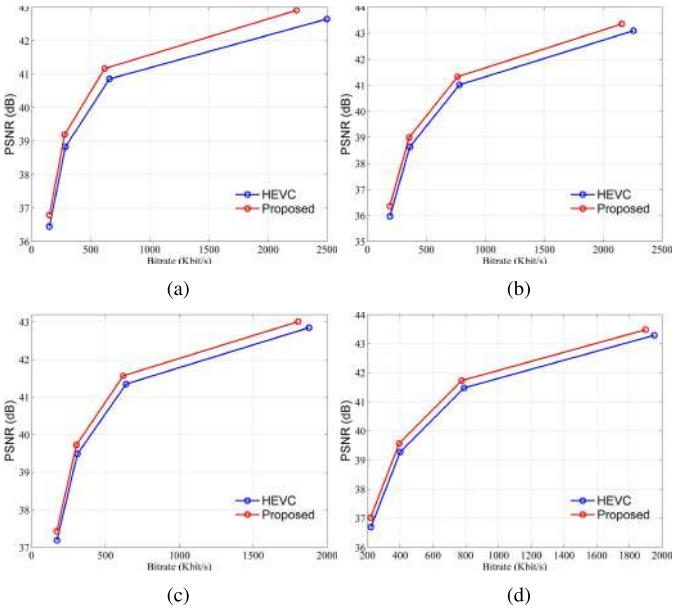


Fig. 7. Luminance PSNR curve for different sequences compressed by HEVC with and without our proposed NALF under LD and RA coding configurations, and ALF is OFF. (a) *Johnny* with LD coding. (b) *KristenAndSara* with LD coding. (c) *Johnny* with RA coding. (d) *KristenAndSara* with RA coding.

respectively. We can see that coding performance is significantly improved in a large bit range with our proposed NALF.

Fig. 8 shows the subjective results for sequences, *Johnny*, *KristenAndSara*, and *Video3*. The decoded images in Fig. 8(a), (e), and (i) are reconstructed without any in-loop filters, the images in Fig. 8(b), (f), and (j) are reconstructed only with DF, the images in Fig. 8(c), (g), and (k) are reconstructed with DF, SAO, and ALF, and the images in Fig. 8(d), (h), and (l) are reconstructed with the proposed NALF and all of the HEVC in-loop filters. From Fig. 8, we can see that DF is efficient only at removing blocking artifacts, e.g., the smooth area of the shirt in *Johnny*. Although SAO and ALF are intended to deal with different types of compression noise, e.g., blocking artifacts, ringing artifacts, mosquito noise, and so on, their performances are limited by the relative large overhead for offset values and filter coefficients. Our proposed filter can efficiently remove different kinds of compression noise, and it can also recover destroyed structures by utilizing nonlocal similar image patches, e.g., most of the lines in the shirt being well recovered in Fig. 8(d).

We also evaluate the computational complexity by comparing the running time for encoder and decoder, respectively. The executable files are compiled by Microsoft Visual Studio 2010, 64 b. The tests are run on Windows 7 operating system with 64 b, and the CPU in the test is Intel Xeon CPU E5-2670 0@2.60 GHz. Compared with HM7.0 encoding, the increment of encoding time due to our NALF is 347% for AI, and 106% and 106% for LD and RA, respectively. Compared with HM7.0 decoding, the increment of decoding time is more than 100 times on average, as shown in the last two rows of Table X. Although the computational complexity is high for practical applications based on our current implementation, there is much room for speedup of the proposed NALF. First, we calculate the percentage of running time for the main modules of the proposed NALF, which is shown in Fig. 9. SVD and

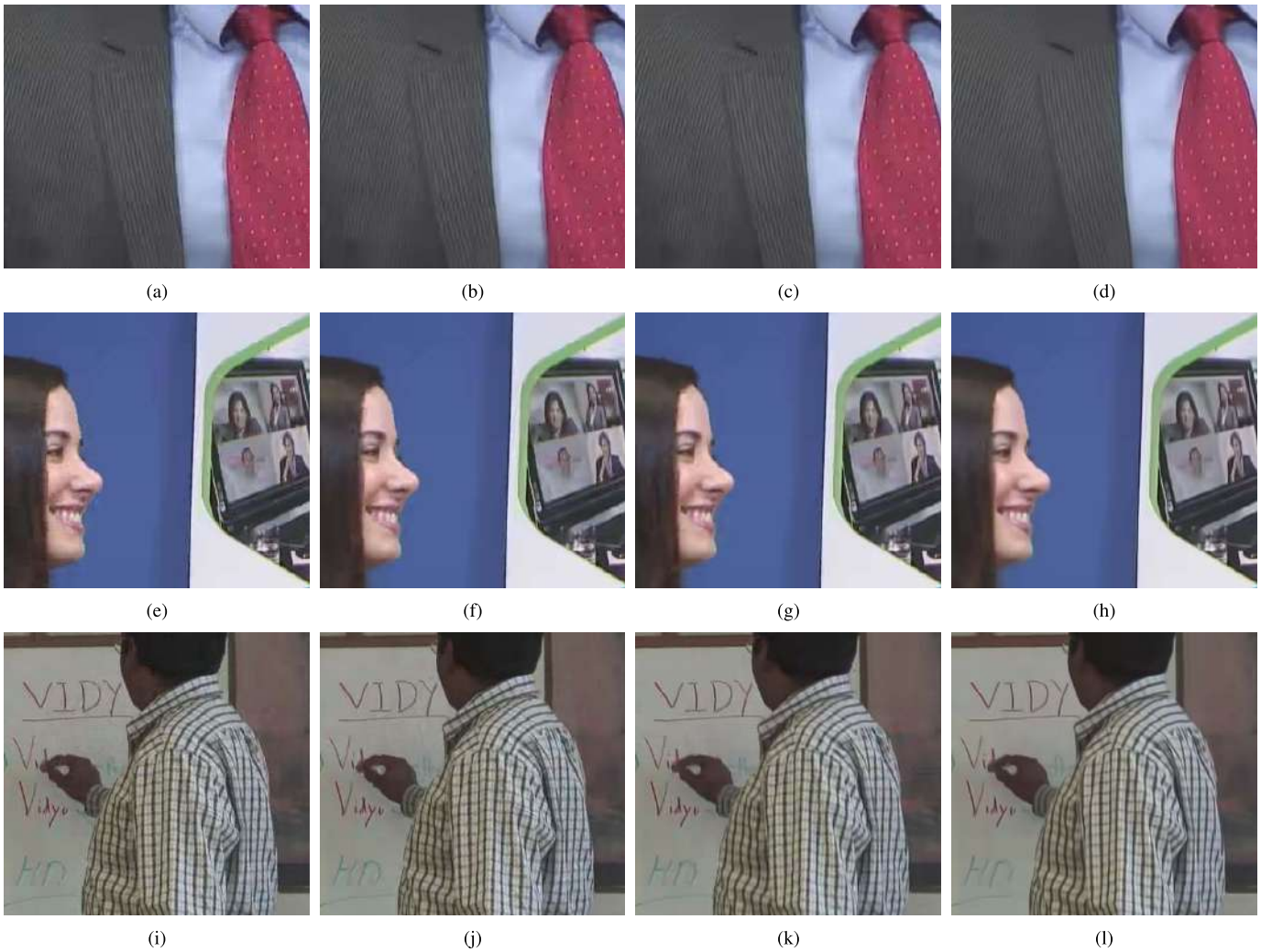


Fig. 8. Subjective image quality comparison, images in the first column reconstructed without any in-loop filters, images in the second column reconstructed with DF, images in the third column reconstructed with DF, SAO and ALF, and images in the last column reconstructed with the proposed NALF and HEVC in-loop filters.

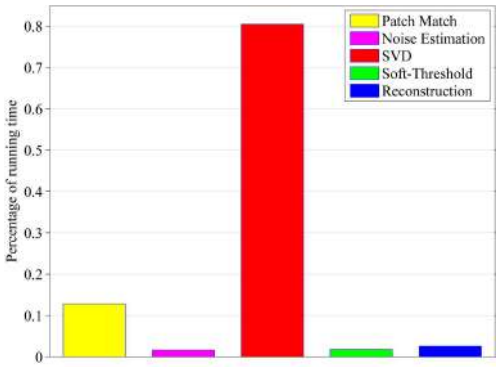


Fig. 9. Percentage of running time for the main modules in the proposed NALF.

Patch Match (searching KNNs) are the most time consuming modules. Herein, the computation of SVD takes up more than 80% running time of NALF, which is implemented with the traditional iterative QR algorithm [37] in our implementation. Some fast algorithms have been proposed for the calculation of singular values in the literature [38], which can further reduce its computational complexity if implemented in the proposed work. In Patch Match module, we use the exhaustive search to find KNNs, which can also be significantly simplified with fast algorithms [39]. Second, the filtering operations of NALF are group-independent for Patch Match, Noise Estimation, SVD, and Soft-Threshold modules, and this makes parallel implementation possible. For a Wide Video Graphics Array (WVGA) sequence, there are more than 20 000 patch groups in one color image with patch size $p = 6$, and neighbor distance $s = 5$. If these image patch groups are all processed in parallel, the running time of NALF will be reduced to 1/20 000 (it is based on running time reduction in linear, but the running time reduction may be a bit smaller than that in practice). Therefore, we think it is possible for real-time video coding applications with the proposed NALF. In this paper, we mainly focus on the performance of NALF, and we will explore fast algorithms for the proposed NALF as our future work, and speed it up by implementing it with fast algorithms and in parallel, e.g., using Graphics processing unit (GPU).

V. Conclusion

In this paper, we have proposed an NALF based on image low-rank prior for video compression. Every video picture is divided into different image patch groups based on their similarity. Soft-thresholding operation is utilized to solve the proposed compression noise reduction problem. We propose an efficient method to derive content-dependent thresholds for every image patch group according to image content characteristics, QPs, and coding modes. We also explore the appropriate position to integrate our proposed NALF in the current HEVC framework and the filter ON/OFF control mechanism at different levels to achieve better coding performance. Experimental results show that the proposed NALF can further improve the compression performance for HEVC, especially for video sequences with lots of similar structure patterns. As one of the first attempts on the coding performance improvement utilizing image nonlocal correlation, the proposed method also has the computational complexity issue that should be improved in the future. This proposes new challenges to the loop filter research with image nonlocal correlations and opens up new space for future exploration.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, 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.

[2] G. J. Sullivan, J.-R. 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.

[3] R. Xiong, F. Wu, J. Xu, X. Fan, C. Luo, and W. Gao, "Analysis of decorrelation transform gain for uncoded wireless image and video communication," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1820–1833, Apr. 2016.

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

[5] 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.

[6] 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.

[7] M. Karczewicz, P. Chen, R. Joshi, X. Wang, W.-J. Chien, and R. Panchal, *Video Coding Technology Proposal by Qualcomm Inc*, JCTVC-A121, ITU-T SG16, Dresden, Germany, Apr. 2010.

[8] C.-Y. Chen *et al.*, "The adaptive loop filtering techniques in the HEVC standard," *Proc. SPIE*, vol. 8499, p. 849913, Oct. 2012.

[9] 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.

[10] S. Wenger, J. Boyce, Y.-W. Huang, C.-Y. Tsai, P. Wu, and M. Li, *Adaptation Parameter Set (APS)*, JCTVC-F747, ITU-T SG16, Turin, Italy, Jul. 2011.

[11] 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.

[12] M. Matsumura, Y. Bandoh, S. Takamura, and H. Jozawa, *In-Loop Filter Based on Non-Local Means Filter*, JCTVC-E206, ITU-T SG16, Geneva, Switzerland, Mar. 2011.

[13] M. Matsumura, S. Takamura, and A. Shimizu, *LCU-Based Framework With Zero Pixel Line Buffers for Non-Local Means Filter*, JCTVC-J0165, ITU-T SG16, Stockholm, Sweden, Jul. 2012.

[14] Q. Han, R. Zhang, W.-K. Cham, and Y. Liu, "Quadtree-based nonlocal Kuan's filtering in video compression," *J. Vis. Commun. Image Represent.*, vol. 25, no. 5, pp. 1044–1055, Jul. 2014.

[15] X. Zhang, W. Lin, S. Wang, and S. Ma, "Nonlocal adaptive in-loop filter via content-dependent soft-thresholding for HEVC," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2015, pp. 465–470.

[16] 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.

[17] 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.

[18] X. Liu, D. Zhai, D. Zhao, G. Zhai, and W. Gao, "Progressive image denoising through hybrid graph Laplacian regularization: A unified framework," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1491–1503, Apr. 2014.

[19] 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.

[20] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.

[21] X. Zhang, R. Xiong, W. Lin, S. Ma, J. Liu, and W. Gao, "Video compression artifact reduction via spatio-temporal multi-hypothesis prediction," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 6048–6061, Dec. 2015.

[22] J. Zhang, R. Xiong, C. Zhao, Y. Zhang, S. Ma, and W. Gao, "CONCOLOR: Constrained non-convex low-rank model for image deblocking," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1246–1259, Mar. 2016.

[23] H. Liu, R. Xiong, X. Zhang, Y. Zhang, S. Ma, and W. Gao, "Nonlocal gradient sparsity regularization for image restoration," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[24] Q. Guo, C. Zhang, Y. Zhang, and H. Liu, "An efficient SVD-based method for image denoising," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 868–880, May 2016.

[25] J. Ren, J. Liu, M. Li, W. Bai, and Z. Guo, "Image blocking artifacts reduction via patch clustering and low-rank minimization," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2013, p. 516.

[26] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, "Compressive sensing via nonlocal low-rank regularization," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3618–3632, Aug. 2014.

[27] X. Zhang, W. Lin, J. Liu, and S. Ma, "Compression noise estimation and reduction via patch clustering," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Dec. 2015, pp. 715–718.

[28] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, Apr. 2009.

[29] C.-J. Hsieh and P. A. Olsen, "Nuclear norm minimization via active subspace selection," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 575–583.

[30] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1532–1546, Sep. 2000.

[31] K. Mohan and M. Fazel, "Iterative reweighted algorithms for matrix rank minimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 3441–3473, Nov. 2012.

[32] J.-S. Lee, "Refined filtering of image noise using local statistics," *Comput. Graph. Image Process.*, vol. 15, no. 4, pp. 380–389, Apr. 1981.

[33] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[34] W. Liu and W. Lin, "Additive white Gaussian noise level estimation in SVD domain for images," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 872–883, Mar. 2013.

[35] F. Bossen, *Common HM Test Conditions and Software Reference Configurations*, JCTVC-I1100, ITU-T SG16, Geneva, Switzerland, Apr. 2012.

[36] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, VCEG-M33, 13th VCEG Meeting, Austin, TX, USA, Apr. 2001.

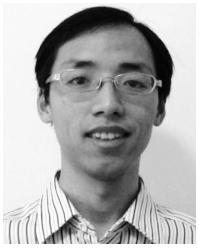
[37] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2012.

[38] A. K. Menon and C. Elkan, "Fast algorithms for approximating the singular value decomposition," *ACM Trans. Knowl. Discovery Data*, vol. 5, no. 2, Feb. 2011, Art. no. 13.

[39] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," in *Proc. ACM SIGGRAPH (SIGGRAPH)*, 2009, Art. no. 24.

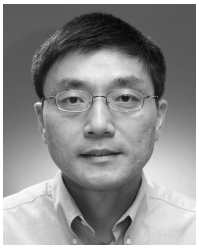
Xinfeng Zhang (M'16) received the B.S. degree in computer science from Hebei University of Technology, Tianjin, China, in 2007, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2014.

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



Ruiqin Xiong (M'08) received the B.S. degree from 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, Beijing, from 2002 to 2007, and a Senior Research Associate with the University of New South Wales, Sydney, NSW, Australia, from 2007 to 2009. He joined Peking University, Beijing, in 2010. His research interests include statistical image modeling and sparse representation, image and video processing, video compression, and multimedia communication.



Weisi Lin (M'92–SM'98–F'16) 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 King's College London, London, U.K., in 1992.

He was involved in teaching and research with Zhongshan University, Shantou University, Shantou, China; the University of Bath, Bath, U.K.; the National University of Singapore, Singapore, the 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 research interests include image processing, perceptual modeling, video compression, multimedia communication, and computer vision.

Jian Zhang (M'14) received the B.Sc. degree from the Department of Mathematics, Harbin Institute of Technology (HIT), Harbin, China, in 2007, and the M.Eng. and Ph.D. degrees from the School of Computer Science and Technology, HIT, in 2009 and 2014, respectively.

He is currently a Post-Doctoral Fellow with the Institute of Digital Media, Peking University, Beijing, China. His research interests include image/video compression and restoration, compressive sensing, sparse representation, and dictionary learning.

Dr. Zhang received the Best Paper Award and Best Student Paper Award at the IEEE International Conference on Visual Communication and Image Processing in 2011 and 2015, respectively.

Shiqi Wang (M'15) received the B.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2008, and the Ph.D. degree in computer application technology from Peking University, Beijing, China, in 2014.

He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently with the Rapid-Rich Object Search Laboratory, Nanyang Technological University, Singapore, as a Research Fellow. His research interests include video compression, image/video quality assessment, and image/video search and analysis.

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 held a Post-Doctoral position with the University of Southern California, Los Angeles, CA, USA, from 2005 to 2007. Then, he joined the Department of Electrical Engineering and Computer Science, Institute of Digital Media, Peking University, Beijing, where he is currently an Associate Professor. He has authored more than 100 technical articles in refereed journals and proceedings in the areas of image and video coding, video processing, video streaming, and transmission.

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

He was a Professor of Computer Science with Harbin Institute of Technology, Harbin, China, from 1991 to 1995, and a Professor 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 published extensively, including five books and more than 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interfaces, and bioinformatics.