Nonlocal In-Loop Filter: The Way Toward Next-Generation Video Coding?

Siwei Ma
Peking University

Xinfeng Zhang
Nanyang Technological University

Jian Zhang, Chuanmin Jia, Shiqi Wang, and Wen Gao
Peking University

Existing in-loop filters rely only on an image's local correlations, largely ignoring nonlocal similarities. The proposed approach uses group-based sparse representation to jointly exploit local and nonlocal self-similarities, laying a novel and meaningful groundwork for in-loop filter design.

High Efficiency Video Coding (HEVC)[1] is the latest video coding standard jointly developed by the International Telecommunication Union–Telecommunication (ITU-T) Video Coding Experts Group (VCEG) and Moving Picture Experts Group (MPEG). Compared to H.264/AVC, HEVC claims to potentially achieve a more than 50 percent coding gain. The in-loop filtering is an important video coding module for improving compression performance by reducing compression artifacts and providing a high-quality reference for subsequent video frames. During the development of HEVC, researchers intensively investigated the performance of three kinds of in-loop filters—the deblocking filter,[2] Sample Adaptive Offset (SAO),[3] and Adaptive Loop Filter (ALF)[4]—and eventually adopted the first two. However, these in-loop filters only take advantage of the image's local correlations, which limits their performance.

Here, we explore the performance of in-loop filters for HEVC by taking advantage of both local and nonlocal correlations in images. We incorporate a nonlocal similarity-based loop filter (NLSLF) into the HEVC standard by simultaneously enforcing the intrinsic local sparsity and nonlocal self-similarity of each frame in the video sequence. For a reconstructed video frame from a previous stage, we first divide it into overlapped image patches and subsequently classify them into different groups based on their similarities. Because these image patches in the same group have similar structures, they can be represented sparsely in a group unit rather than a block unit.[5] We can then reduce the compression artifacts by thresholding the singular values of image patches group by group, based on the sparse property of similar image patches. We also explore two kinds of thresholding methods—hard and soft thresholding—and their related adaptive threshold determination methods. Our extensive experiments on HEVC common test sequences demonstrate that the nonlocal similarity-based in-loop filter significantly improves the compression performance of HEVC, achieving up to an 8.1 percent bitrate savings.

In-Loop Filtering

The deblocking filter was the first adopted in-loop filter in H.264/AVC to reduce the blocking artifacts caused by coarse quantization and motion compensated prediction.[6] Figure 1 shows a typical example of the block boundary with the blocking artifact. H.264/AVC defines a set of low pass filters with different filtering strengths that are applied to 4 × 4 block boundaries. H.264/AVC has five levels of filtering strength, and the filter strength for each block boundary is jointly determined by the quantization parameters, correlations of samples on both side of block boundaries, and the prediction modes (intra- and interprediction).

The deblocking filter in HEVC is similar to that in H.264/AVC. However, in HEVC, it's applied only to 8 × 8 block boundaries, which are the boundaries of coding units (CU), prediction units (PU), or transform units (TU). Due to

1070-986X/16/$33.00 © 2016 IEEE

HEVC's improved prediction accuracy, only three filtering strengths are used, thus reducing complexity compared to H.264/AVC.

SAO is a completely new in-loop filter adopted in HEVC. In contrast to the deblocking filter, which reconstructs only the samples on block boundaries, SAO processes all samples. Because the sizes of coding, prediction, and transform units have been largely extended compared with previous coding standards—that is, the coding unit has been extended from 8 × 8 to 64 × 64, the prediction unit from 4 × 4 to 64 × 64, and the transform unit from 4 × 4 to 32 × 32—the compression artifacts inside the coding blocks can no longer be compensated by the deblocking filter. Therefore, SAO is applied to all samples reconstructed from the deblocking filter by adding an offset to each sample to reduce the distortion.

SAO has proven to be a powerful tool to reduce ringing and contouring artifacts. To adapt the image content, SAO first divides a reconstructed picture into different regions and then derives an optimal offset for each region by minimizing the distortion between the original and reconstructed samples. SAO can use different offsets sample by sample in a region, depending on the sample classification strategy. In HEVC, two SAO types were adopted: edge offset and band offset. For the edge offset, the sample classification is based on comparing the current and the neighboring samples according to four one-dimensional neighboring patterns (see Figure 2). For the band offset, the sample classification is based on sample values, and the sample value range is equally divided into 32 bands. These offset values and region indices are signaled in the bitstream, which can impose a relatively large overhead.

ALF is a Wiener-based adaptive filter; its coefficients are derived by minimizing the mean square errors between original and reconstructed samples. Numerous recent efforts have been dedicated to developing high-efficiency and low-complexity ALF approaches. In HEVC reference software HM7.0, the filter shape of ALF is a combination of a 9 × 7-tap cross shape and a 3 × 3-tap rectangular shape, as Figure 3 illustrates. Therefore, only correlations within a local patch are used to reduce the compression artifacts.

To adapt the properties of an input frame, up to 16 filters are derived for different regions of the luminance component. Such high adaptability also creates a large overhead, which should be signaled in the bitstream. Therefore, these regions must be merged at the encoder side based on rate-distortion optimization (RDO), which makes neighboring regions share the same filters to achieve a good tradeoff between the filter performance and overheads. One of us (Zhang) and colleagues proposed reusing the filter coefficients and regions division in the previous encoded frame to reduce overheads.[7] Stephan Wenger and his colleagues proposed placing the filter coefficient parameters in a picture-level header called the *Adaptation Parameter Set* (APS), which

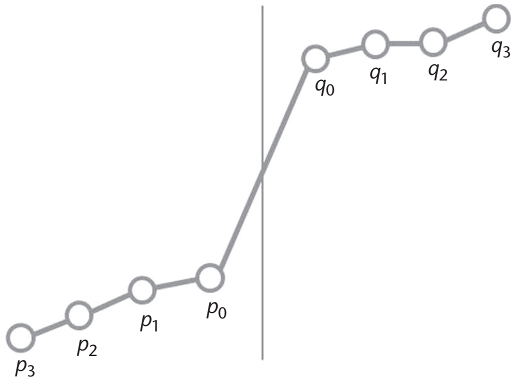

Figure 1. A one-dimensional example of the block boundary with the blocking artifact. Here, $\{p_i\}$ and $\{q_i\}$ are pixels in neighboring blocks.

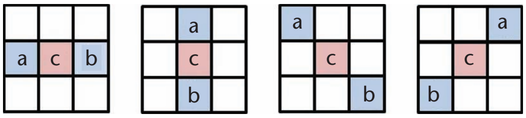

Figure 2. Four 1D directional patterns for edge offset sample classification. The samples in the positions, a, b and c, are used for comparison.

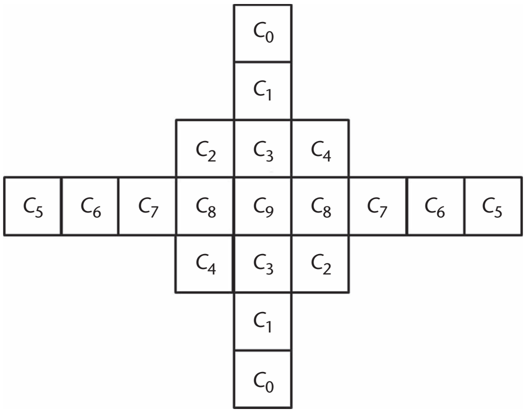

Figure 3. Adaptive loop filter (ALF) shape in HM7.0 (each square corresponds to a sample). The notations, $c_1, c_2, \ldots, c_9$ are the filter coefficients.

Related Work in Nonlocal Image Filters

In existing video coding standards, in-loop filters focus only on the local correlation within image patches without fully considering nonlocal similarities. However, in image restoration and denoising fields, researchers have proposed many methods based on image nonlocal similarities.[1–5]

Antoni Buades and his colleagues proposed the famous nonlocal means filter (NLM) to remove different kinds of noise by predicting each pixel with a weighted average of nonlocal pixels, where the weights are determined by the similarity of image patches located at the source and target coordinates.[1] The well-known block-matching and 3D filtering (BM3D) denoising filter stacks nonlocal similar image patches into 3D matrices and removes noise by shrinking coefficients of 3D transform of similar image patches based on the image-sparse prior model.[2] Other research used the nonlocal similar image patches to suppress compression artifacts, which is achieved by adaptively combining the pixels restored by the NLM filter and reconstructed pixels according to the reliability of NLM prediction and quantization noise in the transform domain.[3–5] In other work, the authors use a group of nonlocal similar image patches to construct image-sparse representation, which can be further applied to image deblurring, denoising, and inpainting.[6–8] Although these nonlocal methods significantly improve the quality of restored images, all of them are treated as post-processing filters and thus don't fully exploit the compression information.

Masaaki Matsumura and his colleagues first introduced the NLM filter to compensate for the shortcomings of HEVC with only image-local prior models; to improve the coding performance, they used delicately designed patch shapes, search window shapes, and optimizing filter on/off control modules.[9,10] Finally, Qinglong Han and his colleagues also employed nonlocal similar image patches in a quadtree-based Kuan's filter to suppress compression artifacts; the pixels restored by the NLM filter and the reconstructed pixels are adaptively combined according to the variance of image signals and quantization noise.[11] However, the weights in these filters are difficult to determine, leading to limited coding performance improvement.

References

1. A. Buades, B. Coll, and J. M. Morel, "A Non-Local Algorithm for Image Denoising," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition* (CVPR), vol. 2, 2005, pp. 60–65.
2. K. Dabov et al., "Image De-Noising by Sparse 3D Transform-Domain Collaborative Filtering," *IEEE Trans. Image Processing*, vol. 16, no. 8, 2007, pp. 2080–2095.
3. X. Zhang et al., "Reducing Blocking Artifacts in Compressed Images via Transform-Domain Non-local Coefficients Estimation," *Proc. IEEE Int'l Conf. Multimedia and Expo* (ICME), 2012, pp. 836–841.
4. X. Zhang et al., "Compression Artifact Reduction by Overlapped-Block Transform Coefficient Estimation with Block Similarity," *IEEE Trans. Image Processing*, vol. 22, no. 12, 2013, pp. 4613–4626.
5. X. Zhang et al., "Artifact Reduction of Compressed Video via Three-Dimensional Adaptive Estimation of Transform Coefficients," *Proc. IEEE Int'l Conf. Image Processing* (ICIP), 2014, pp. 4567–4571.
6. J. Zhang et al., "Image Restoration Using Joint Statistical Modeling in a Space-Transform Domain," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 24, no. 6, 2014, pp. 915–928.
7. X. Zhang et al., "Compression Noise Estimation and Reduction via Patch Clustering," *Proc. Asia-Pacific Signal and Information Processing Assoc. Ann. Summit and Conf.*, vol. 16, no. 19, 2015, pp. 715–718.
8. J. Zhang, D. Zhao, and W. Gao, "Group-Based Sparse Representation for Image Restoration," *IEEE Trans. Image Processing*, vol. 23, no. 8, 2014, pp. 3336–3351.
9. M. Matsumura et al., "In-Loop Filter Based on Non-local Means Filter," *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-E206*, 2011; http://phenix.int-evry.fr/jct/doc_end_user/documents/5_Geneva/wg11.
10. M. Matsumura, S. Takamura, and A. Shimizu, "Largest Coding Unit Based Framework for Non-Local Means Filter," *Proc. Asia-Pacific Signal Information Processing Assoc. Ann. Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, Dec. 2012, pp. 1–4.
11. Q. Han et al., "Quadtree-Based Non-Local Kuans Filtering in Video Compression," *J. Visual Communication and Image Representation*, vol. 25, no. 5, 2014, pp. 1044–1055.

makes in-loop filter parameters reuse more flexible with APS indices.[8]

The Nonlocal Similarity-Based In-Loop Filter

In addition to image local-correlation-based filters, many nonlocal-correlation-based filters have been proposed in the literature (see the "Related Work in Nonlocal Image Filters" sidebar). In our previous work,[5] we formulated a new sparse representation model in terms of a group of similar image patches. Our group-based sparse representation (GSR) model can exploit the local sparsity and the nonlocal self-similarity of natural images simultaneously in a unified framework. Here, we describe how the NLSLF is designed in stages based on the GSR model.

Patch Grouping

The basic idea of GSR is to adaptively sparsify the natural image in the domain of a group. Thus, we first show how to construct a group.

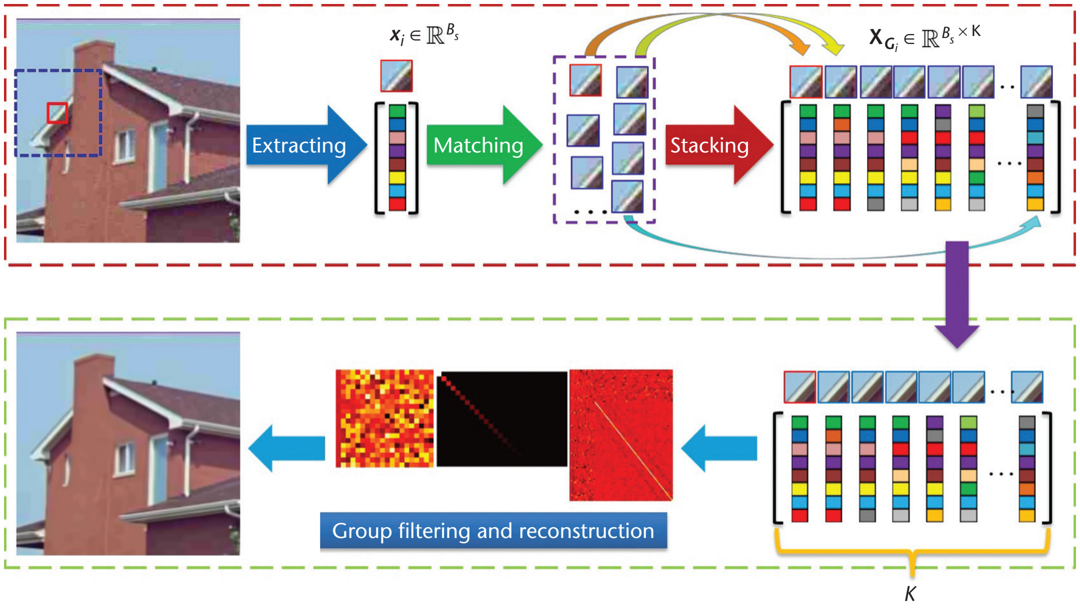
$x_i \in \mathbb{R}^{B_s}$
$\mathbf{X}_{G_i} \in \mathbb{R}^{B_s \times K}$
Extracting
Matching
Stacking
Group filtering and reconstruction
K

Figure 4. Framework of the nonlocal similarity-based loop filter (NLSLF). The high-quality image is reconstructed via patch grouping, group filtering, and reconstruction.

In fact, each group is represented by a matrix, which is composed of nonlocal patches with similar structures. For a video frame, I, we first divide it into S overlapped image patches with the size of $\sqrt{B_s} \times \sqrt{B_s}$. Each patch is reorganized into a vector, x_k, $k = 1, 2, ..., S$, as illustrated in Figure 4. For every image patch, we find K nearest neighbors according to the Euclidean distance between different image patches,

$$d(\boldsymbol{x}_i, \mathbf{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2. \quad (1)$$

These K similar image patches are stacked into a matrix of size $B_s \times K$,

$$\mathbf{X}_{G_i} = \left[x_{G_i,1}, x_{G_i,2}, ..., x_{G_i,K}\right]. \quad (2)$$

Here, $\mathbf{X}_{G_i}$ contains all the image patches with similar structures, which we call a *group*.

Group Filtering and Reconstruction

Because the image patches in the same group are very similar, they can be represented sparsely. For each group, we apply singular value decomposition (SVD) and get image sparse representation,

$$\mathbf{X}_{G_i} = \mathbf{U}_{G_i}\Sigma_{G_i}\mathbf{V}_{G_i}^T = \sum_{k=1}^{M} \Upsilon_{G_i,k}\left(u_{G_i,k}v_{G_i,k}^T\right), \quad (3)$$

where $\Upsilon_{G_i} = [\Upsilon_{G_i,1}, \Upsilon_{G_i,2}; ..., \Upsilon_{G_i,M}]$ is a column vector, $\Sigma_{G_i} = diag(\Upsilon_{G_i})$ is a diagonal matrix with the elements of $\gamma\ _{G_i}$ as its main diagonal, and $\mathrm{u}_{G_i,k}\mathrm{v}_{G_i,k}^T$ are the columns of $\mathbf{U}_{G_i}$ and V_{G_i}, respectively. M is the maximum dimension of matrix $\mathbf{X}_{G_i}$.

The matrix composed of the corresponding compressed video frame is formulated as

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}, \quad (4)$$

where $\mathbf{N}$ is the compression noise and $\mathbf{X}$ and $\mathbf{Y}$ (without any subscript) represent the original and reconstructed frames, respectively. To derive the sparse representation parameters, we apply thresholding, which is a widely used operation for coefficients with sparse property in image denoising problems. We apply two kinds of the thresholding methods—hard and soft thresholding— to the singular values in Υ_{G_i}, which is composed of singular values of matrix **Y,**

$$\boldsymbol{\alpha}_{G_i}^{(h)} = \text{hard}(\Upsilon_{G_i,\tau}) \text{ and} \quad (5)$$

$$\boldsymbol{\alpha}_{G_i}^{(s)} = \text{soft}(\Upsilon_{G_i,\tau}), \quad (6)$$

where the hard and soft thresholding are defined as

$$\text{hard}\,(\boldsymbol{x}, \tau) = \text{sign}(\boldsymbol{x}) \odot \left(\text{abs}(\boldsymbol{x}) - \tau 1\right) \quad (7)$$

$$\text{soft}\,(\boldsymbol{x}, \tau) = \text{sign}(\boldsymbol{x}) \odot \max\left(\text{abs}(\boldsymbol{x}) - \tau 1, 0\right). \quad (8)$$

Here, $\odot$ stands for the element-wise product of two vectors, *sign*$(\cdot)$ is the function extracting the sign of every element of a vector, 1 is an all-ones vector, and τ denotes the threshold. After achieving the shrunken singular values, the restored group of image patches $\hat{x}$ is given by

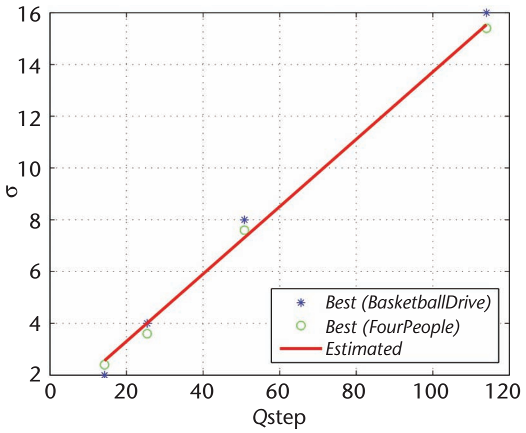

Figure 5. The relationship between Qstep and the standard deviation of compression noise. The linear function can fits well with their relationship.

$$\hat{x} = \sum_{k=1}^{M} \boldsymbol{\alpha}_{G_i,k}(u_{G_i,k} v_{G_i,k}^{T}). \tag{9}$$

Because these image patches are overlap extracted, we simply take the average of the overlapped samples as the final filtered values.

Threshold Estimation

Based on the above discussion, we determine the filtering strength by the thresholding-level parameter τ in Equations 5 and 6. However, given that various video content is compressed with different quantization parameters, this is a nontrivial problem that has not been well resolved. In essence, the optimal threshold is closely related with the standard deviation of noise denoted as σ_n, and larger thresholds correspond to higher σ_n values.

In video coding, the compression noise is mainly caused by quantizing the transform coefficients. Therefore, we can use quantization steps to determine the standard deviation of the compression noise and a scale factor to adapt different prediction modes, including intra- and interprediction.

For hard thresholding, the optimal values of σ_n are derived experimentally based on the sequences *BasketballDrive* and *FourPeople*, compressed with different quantization parameters ($\mathrm{QP} = 27,\ 32,\ 38,\ 45$), which are further converted to the quantization step sizes (Qsteps), as Figure 5 shows. We can infer that different sequences with the same quantization parameter or Qstep have similar optimal values of σ_n, implying that σ_n is closely related with the quantization parameter or Qstep. Inspired by this, we estimate the optimal value of σ_n directly from the Qstep by curve fitting using the following empirical formulation,

$$\sigma = a * \mathrm{Qstep} + b, \tag{10}$$

where the Qstep can be easily derived from the quantization parameter based on the following relationship in HEVC:

$$\mathrm{Qstep} = 2^{\frac{(QP-4)}{6}}. \tag{11}$$

Table 1 shows the parameters (a, b) for different coding configurations.

Based on the filtering performance, we further use the size and number of similar image patches in one group as a scale factor

$$\tau = \sigma_n * (B_s + \sqrt{K}), \tag{12}$$

where σ_n is the standard deviation of compression noise for the whole image, which is estimated based on Equation 10.

For soft thresholding, based on the filtering performance, we take the optimal threshold formulation for generalized Gaussian signals,

$$\tau = \frac{c\sigma_n^2}{\sigma_x}, \tag{13}$$

where σ_x is the standard deviation of original signals that can be estimated by

$$\sigma_x^2 = \sigma_y^2 - \sigma_n^2. \tag{14}$$

Because the variance of compression noise, σ_n, is derived at the encoder side, we quantize it into the nearest integer range,[9] which is signaled with 4 bits and transmitted in the bitstream. Therefore, 12 bits are encoded in total for one frame with three color components—for example, YUV. The two thresholds for both

Table 1. The coefficient for estimating σ for all configurations.

	All intra coding		Low delay B coding		Random access coding	
Color component	a	b	a	b	a	b
Y	0.13000	0.7100	0.10450	0.4870	0.10450	0.4870
U	0.06623	0.8617	0.03771	0.8833	0.03771	0.8833
V	0.06623	0.8617	0.03771	0.8833	0.03771	0.8833

hard and soft thresholding operations increase with the standard deviation of compression noise, which implies that the frames with more noise should be filtered with higher strength. Furthermore, the thresholds decrease with the standard deviation of signals, which can avoid over-smoothing for smooth areas.

Hard and soft thresholding operations increase with the standard deviation of compression noise, which implies that the frames with more noise should be filtered with higher strength.

Filtering On/Off Control

To ensure that the NLSLF consistently leads to distortion reduction, we introduce on/off control flags for frame and largest coding unit (LCU) levels, which should be signaled in the bitstream. Specifically, regarding the frame-level on/off control, three flags—*Filtered_Y, Filtered_U,* and *Filtered_V*—are designed for the corresponding color components Y, U, and V, respectively. When the distortions of the filtered image decrease, the corresponding flag signals as *true,* indicating that the image color component is finally filtered. For the on/off control at the LCU level, each LCU needs only one flag *Filterd_LCU*[*i*] to indicate the on/off filtering for the luminance component of the corresponding LCU. In the picture header syntax structure, three bits are encoded to signal frame-level control flags for each color component, respectively. We place the syntax elements of the LCU-level control flags in coding tree unit parts, using only one bit for each LCU.

Experimental Results and Analysis

In our experiments, we implement the nonlocal similarity-based in-loop filter in the HEVC reference software, HM12.0. We denote the hard-threshold filtering (with the threshold in Equation 12) as NLSLF-H, and the soft-threshold filtering (with the threshold in Equation 13) as NLSLF-S. To better analyze the performance of the nonlocal similarity-based in-loop filter, we further integrate the ALF from HM3.0 into HM12.0 (in which the ALF tool has been removed) and compare the nonlocal similarity-based in-loop filter with ALF.

The test video sequences in our experiments are widely used in HEVC common test conditions. There are 20 test sequences that are classified into six categories (Classes A–F). The resolutions for the first five categories are as follows:

- Class A: 2560 × 1600,
- Class B: 1920 × 1080,
- Class C: 832 × 480,
- Class D: 416 × 240, and
- Class E is 1280 × 720.

Class F contains screen videos with three different resolutions: 1280 × 720, 1024 × 768, and 832 × 480.

We tested four typical quantization parameters—22, 27, 32, and 37—and three common coding configurations: all intra coding (AI), low delay B (LDB) coding, and random access (RA) coding. Along with the increase of K and B_s, the computational complexity increases rapidly, while the filtering performance might decrease for some sequences because dissimilar structures are more likely to be included. Therefore, in our experiments, the size of image patches is set to $B_s = 6$, and the number of nearest neighbors for each image patch is set to $K = 30$ for all the sequences. For each frame, we extract image patches every five pixels according to the raster scanning order, which makes the image patches overlap.

First, we treat the HM12.0 with and without ALF as anchors. The overall coding performances of NLSLF-S and NLSLF-H with only frame-level control are shown in Tables 2–5. Both of the two thresholding filters with nonlocal image patches achieve significant bitrate savings compared to HM12.0 without ALF. NLSLF-S achieves 3.2 percent, 3.1 percent, and 4.0 percent bitrate savings on average for the AI, LDB, and RA configurations, respectively. Moreover, NLSLF-H achieves 4.1 percent, 3.3 percent, and 4.4 percent bitrate savings on

Table 2. Performance of the nonlocal similarity-based loop filter with soft thresholding (NLSLF-S) on HM12.0 with adaptive loop filtering (ALF) turned off.

	All intra coding (%)			Low delay B (LDB) coding (%)			Random access coding (%)		
Sequences	Y	U	V	Y	U	V	Y	U	V
Class A	−4.3	−4.0	−3.9	−3.5	−3.3	−2.3	−4.8	−6.1	−5.7
Class B	−2.9	−3.3	−4.0	−3.0	−4.2	−4.2	−4.3	−5.5	−4.7
Class C	−2.8	−4.6	−6.2	−1.6	−3.4	−5.4	−2.1	−5.1	−6.5
Class D	−2.0	−4.5	−5.5	−1.3	−2.4	−2.5	−1.6	−3.5	−4.4
Class E	−5.8	−5.3	−4.4	−7.9	−10.0	−9.5	−9.8	−9.4	−8.6
Class F	−2.5	−3.1	−3.4	−1.7	−2.8	−3.3	−2.2	−4.4	−4.7
Overall	−3.4	−4.1	−4.6	−3.2	−4.4	−4.5	−4.1	−5.6	−5.8

Table 3. Performance of the nonlocal similarity-based loop filter with soft thresholding (NLSLF-S) on HM12.0 with adaptive loop filtering (ALF) turned on.

	All intra coding (%)			Low delay B (LDB) coding (%)			Random access coding (%)		
Sequences	Y	U	V	Y	U	V	Y	U	V
Class A	−1.8	−2.3	−2.4	−1.0	−3.9	−2.5	−2.2	−5.2	−5.0
Class B	−1.8	−2.1	−3.0	−1.8	−3.9	−4.7	−2.6	−5.0	−5.2
Class C	−2.7	−3.5	−4.5	−1.7	−4.4	−5.9	−2.2	−5.6	−6.4
Class D	−1.9	−2.8	−3.7	−1.7	−2.2	−3.2	−1.8	−3.7	−4.6
Class E	−3.9	−2.8	−2.1	−6.1	−7.5	−6.0	−7.4	−7.3	−6.2
Class F	−2.4	−2.9	−3.2	−1.9	−3.6	−3.9	−2.0	−4.2	−4.5
Overall	−2.4	−2.7	−3.2	−2.4	−4.2	−4.4	−3.0	−5.1	−5.3

Table 4. Performance of the nonlocal similarity-based loop filter with hard thresholding (NLSLF-H) on HM12.0 with adaptive loop filtering (ALF) turned off.

	All intra coding (%)			Low delay B (LDB) coding (%)			Random access coding (%)		
Sequences	Y	U	V	Y	U	V	Y	U	V
Class A	−4.9	−3.0	−3.5	−3.1	−1.2	−1.4	−4.2	−3.1	−2.8
Class B	−3.2	−2.2	−3.9	−3.2	−3.5	−3.7	−4.3	−3.9	−3.8
Class C	−3.6	−4.9	−6.9	−1.9	−3.4	−4.8	−2.5	−4.2	−5.9
Class D	−3.1	−4.4	−5.9	−1.5	−2.5	−2.8	−2.1	−3.4	−3.4
Class E	−7.1	−8.5	−8.9	−7.4	−9.5	−10.5	−10.0	−11.4	−12.1
Class F	−3.5	−4.4	−5.0	−2.4	−2.8	−3.6	−3.0	−5.0	−5.4
Overall	−4.2	−4.6	−5.7	−3.3	−3.8	−4.5	−4.3	−5.2	−5.6

average for the all intra, LDB, and random access configurations, respectively, compared to HM12.0 without ALF. When the nonlocal similarity-based in-loop filters are combined with ALF, NLSLF-S achieves approximately 2.6 percent, 2.6 percent, and 3.2 percent bitrate savings for all intra, LDB, and random access coding, respectively, and NLSLF-H achieves approximately 3.1 percent, 2.8 percent, and 3.4 percent bitrate savings for all intro, LDB, and random access coding, respectively, compared with HM12.0 with ALF.

Although the NLSLF improvements are not as significant as those achieved without ALF, they can still further improve the performance of HEVC with ALF. This verifies that nonlocal

Table 5. Performance of the nonlocal similarity-based loop filter with hard thresholding (NLSLF-H) on HM12.0 with adaptive loop filtering (ALF) turned on.

	All intra coding (%)			Low delay B (LDB) coding (%)			Random access coding (%)		
Sequences	Y	U	V	Y	U	V	Y	U	V
Class A	−2.1	−1.4	−1.8	−1.0	−1.6	−1.3	−1.7	−2.3	−2.1
Class B	−1.9	−1.0	−2.5	−2.1	−2.9	−3.3	−2.6	−3.0	−3.8
Class C	−3.1	−2.6	−5.0	−2.0	−4.0	−5.1	−2.2	−4.3	−5.9
Class D	−2.6	−1.6	−3.1	−1.6	−2.5	−3.0	−1.9	−3.6	−3.8
Class E	−4.9	−4.5	−3.9	−5.5	−5.5	−5.6	−7.5	−7.5	−6.8
Class F	−3.1	−4.3	−5.0	−2.8	−3.5	−3.6	−2.9	−4.7	−5.3
Overall	−2.9	−2.6	−3.5	−2.5	−3.3	−3.7	−3.1	−4.2	−4.6

Table 6. Performance of the nonlocal similarity-based loop filter with soft thresholding (NLSLF-S) with largest coding unit (LCU) level control for each sequence.

		All intra coding (%)			Low delay B (LDB) coding (%)			Random access coding (%)		
Sequences		Y	U	V	Y	U	V	Y	U	V
Class A	Traffic	−2.0	−2.0	−2.4	−2.3	−1.9	−1.5	−2.9	−3.9	−3.2
	PeopleOnStreet	−2.4	−2.7	−2.4	−2.8	−5.2	−3.4	−2.5	−5.8	−6.1
Class B	Kimono	−1.9	−1.0	−1.8	−3.0	−4.3	−4.4	−1.5	−2.8	−4.1
	ParkScene	−0.6	−0.5	−0.9	−0.9	1.4	0.5	−1.3	−0.4	−0.1
	Cactus	−2.4	−1.5	−4.5	−4.1	−2.3	−4.9	−4.3	−6.8	−7.3
	BasketballDrive	−1.9	−4.7	−5.2	−2.5	−9.1	−8.5	−2.3	−8.0	−6.9
	BQTerrace	−2.8	−2.5	−2.7	−4.6	−2.5	−4.9	−7.2	−4.4	−5.6
Class C	BasketballDrill	−4.3	−7.0	−8.6	−3.1	−10.2	−11.9	−3.3	−11.8	−13.0
	BQMall	−4.2	−3.8	−4.0	−4.7	−4.3	−4.5	−4.4	−5.4	−5.0
	PartyScene	−0.9	−1.3	−1.8	−1.4	0.9	1.5	−1.8	−0.1	−0.2
	RaceHorsesC	−1.3	−1.8	−3.6	−2.7	−3.1	−7.6	−2.6	−3.6	−7.3
Class D	BasketballPass	−3.4	−4.5	−4.7	−2.4	−4.0	−3.6	−2.0	−5.2	−4.6
	BQSquare	−1.7	−0.9	−2.6	−1.5	1.0	−0.4	−2.4	−0.8	−1.9
	BlowingBubbles	−1.1	−2.9	−3.6	−1.9	−2.7	−0.5	−2.2	−3.7	−4.1
	RaceHorses	−2.1	−3.3	−4.4	−3.3	−1.0	−5.6	−2.7	−4.6	−7.2
Class E	FourPeople	−3.2	−2.5	−1.7	−4.8	−5.6	−4.5	−5.6	−5.2	−4.7
	Johnny	−4.9	−3.0	−1.7	−6.7	−7.7	−5.3	−8.1	−6.8	−5.8
	KristenAndSara	−3.6	−2.6	−2.7	−5.2	−5.0	−4.4	−6.0	−7.4	−5.2
Class F	BasketballDrillText	−4.4	−6.7	−7.8	−3.3	−8.2	−8.5	−3.7	−10.3	−10.8
	ChinaSpeed	−1.7	−2.5	−2.5	−2.9	−2.1	−3.1	−2.3	−4.6	−4.4
	SlideEditing	−1.9	−0.5	−0.8	−2.1	−0.2	−0.4	−2.1	−0.5	−0.8
	SlideShow	−1.4	−1.5	−1.4	−0.8	−3.2	−1.4	0.0	−0.7	−0.9
Overall		−2.5	−2.7	−3.1	−3.1	−3.7	−3.9	−3.3	−4.8	−5.0

similarity offers more benefits for compression artifact reduction than local similarity alone. Because hard- and soft-thresholding operations are suitable for signals with different distributions, they show different coding gains on different sequences. Although NLSLF-H achieves better performance for most sequences than NLSLF-S in our experiments, soft thresholding outperforms hard thresholding for some sequences, such as for Class E in the LDB coding configuration and Class A in the LDB and random access coding configurations.

Table 6 shows the detailed results of NLSLF-S with LCU-level control for each sequence.

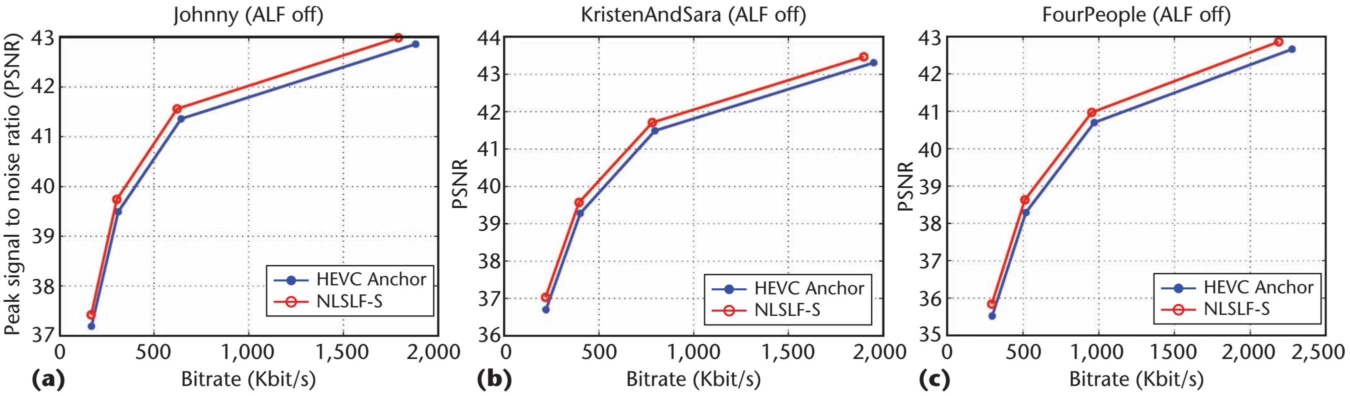

Figure 6. The rate-distortion performance of the nonlocal similarity-based loop filter with soft thresholding (NLSLF-S) compared with HEVC with the adaptive in-loop filter turned off. The test involved three sequences: (a) Johnny, (b) KristenAndSara, and (c) FourPeople. All three sequences are compressed by HEVC RA coding.

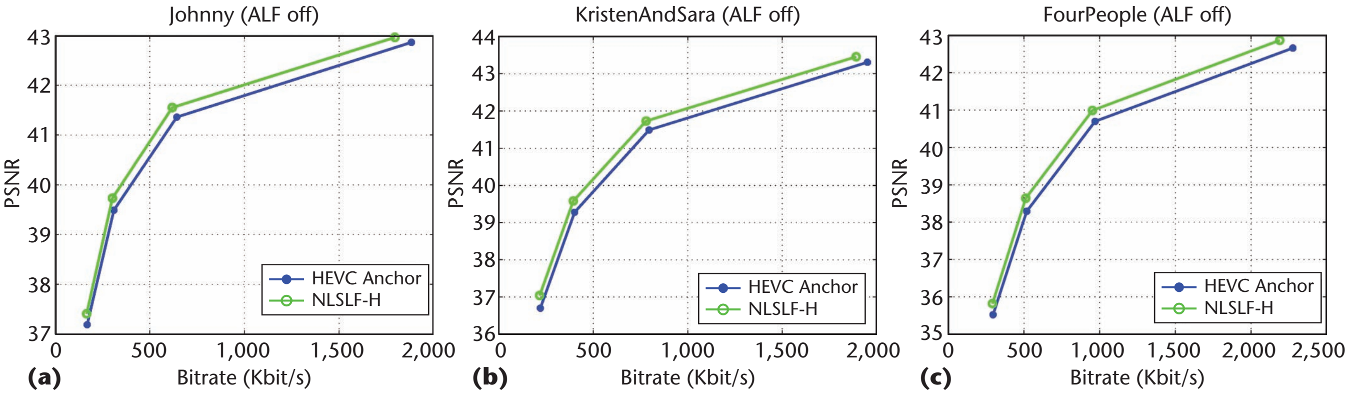

Figure 7. The rate-distortion performance of the nonlocal similarity-based loop filter with hard thresholding (NLSLF-H) compared with HEVC with the adaptive in- loop filter turned off. The test involved three sequences: (a) Johnny, (b) KristenAndSara, and (c) FourPeople. All three sequences are compressed by HEVC RA coding.

Although LCU-level control increases overheads, it can also improve coding efficiency by avoiding the over-smoothing case. Further, this shows that room still exists for improving the filtering efficiency by designing more reasonable thresholds for group-based sparse coefficients. Figures 6 and 7 illustrate the rate-distortion curves of NLSF and HEVC without ALF for the sequences *Johnny, KristenAndSara,* and *FourPeople,* which are compressed at different quantization parameters under the random access configuration. As the figures show, the coding performance is significantly improved in a wide bit range with the nonlocal similarity-based in-loop filters.

We further compare the visual quality of the decoded video frames with different in-loop filters in Figure 8. The deblocking filter removes only the blocking artifacts, and it is difficult to reduce other artifacts, such as the ringing artifacts around the coat's stripes in the *Johnny* image. Although SAO can process all the reconstructed samples, its performance is constrained by the large overheads, such that blurring edges still exist. The nonlocal similarity-based filters can efficiently remove different kinds of compression artifacts, as well as recover destroyed structures by utilizing nonlocal similar image patches, such as recovering most of the lines in *Johnny*'s coat.

Although NLSLF achieves significant improvement for video coding, it also introduces many computational burdens, especially due to SVD. Compared with HM12.0 encoding, NLSLF-H's encoding time increase is 133 percent, 30 percent, and 33 percent for all intra, LDB, and random access coding, respectively. This also proposes new challenges for loop filter research

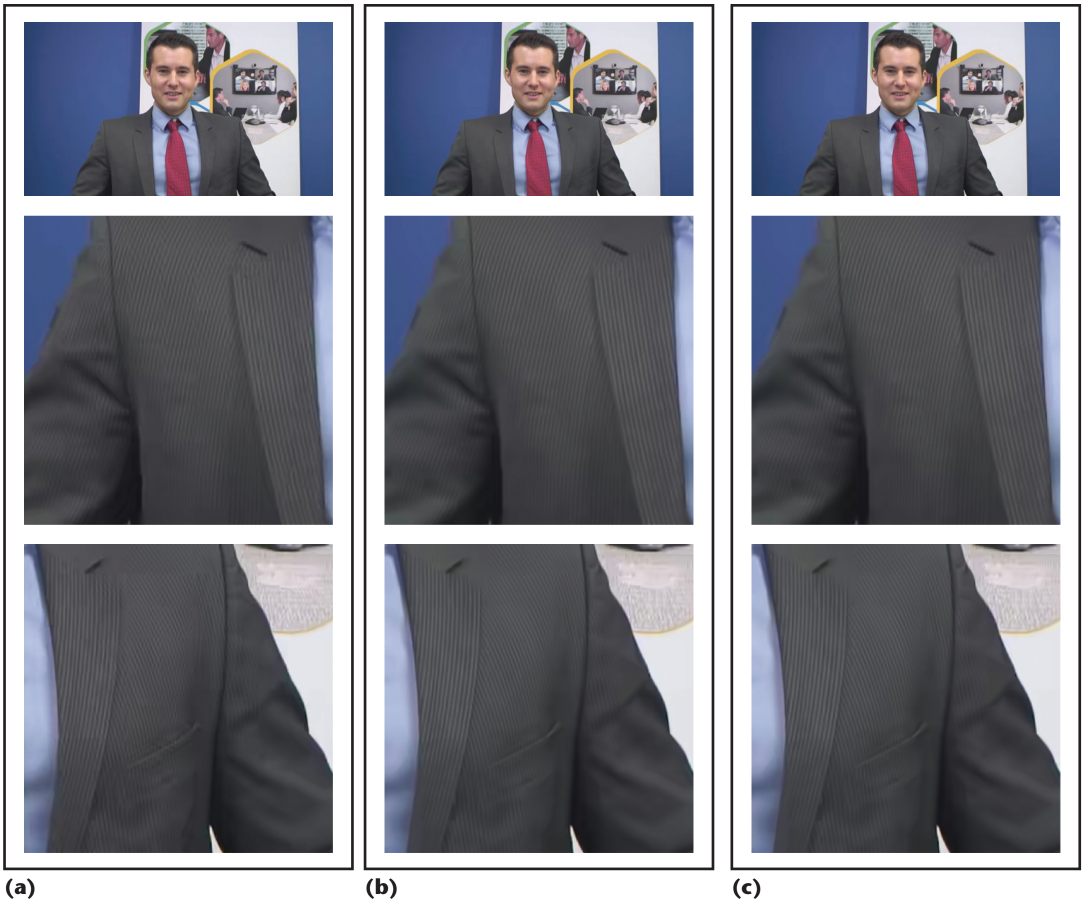

Figure 8. Visual quality comparison for the Johnny sequence when the adaptive in-loop filter (ALF) is off: (a) images reconstructed with HEVC Anchor; (b) Images reconstructed with the nonlocal similarity-based loop filter with soft thresholding (NLSLF-S), and (c) images reconstructed with NLSLF with hard thresholding (NLSLF-H).

on image nonlocal correlations, which we plan to explore in our future work.

The novelty in our approach lies in adopting the nonlocal model in the in-loop filtering process, which leads to reconstructed frames with higher fidelity. To estimate the noise level, we examined different kinds of thresholding operations, confirming that the nonlocal strategy can significantly improve the coding efficiency. This offers new opportunities for in-loop filter research with nonlocal prior models. It also opens up new space for future exploration in nonlocal-inspired high-efficiency video compression.

Apart from in-loop filtering, the nonlocal information can motivate the design of other key modules in video compression as well. Traditional video coding technologies focus mainly on reducing the local redundancies by intraprediction with limited neighboring samples. This interprediction can be regarded as a simplified version of nonlocal prediction, which obtains predictions from a relatively large range compared to intraprediction, leading to significant performance improvement.

However, to the maximum extent, only a unique pair of patches can be employed, such as one image patch in unidirectional predictions and two image patches in bidirectional predictions. This significantly limits the prediction technique's potential, as the number of similar image patches can be further extended to fully exploit the spatial and temporal redundancies. With the new technological advances in hardware and software, we could have foreseen the arrival and maturity of these nonlocal-based coding techniques. We also believe that the nonlocal-based video coding technology described in this article—or similar technologies developed based on it—could play an important role in the future of video standardization. **MM**

Acknowledgments

This special issue is a collaboration between the 2015 IEEE International Symposium on

Multimedia (ISM 2015) and *IEEE MultiMedia*. This article is an extended version of "Non-Local Structure-Based Filter for Video Coding," presented at ISM 2015. This work was supported in part by the National Natural Science Foundation of China (grants 61322106, 61572047, and 61571017) and the National Basic Research Program of China 973 Program (grant 2015CB351800).

References

1. G. Sullivan et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, 2012, pp. 1649–1668.
2. A. Norkin et al., "HEVC De-blocking Filter," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, 2012, pp. 1746–1754.
3. C.-M. Fu et al., "Sample Adaptive Offset in the HEVC Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, 2012, pp. 1755–1764.
4. C.-Y. Tsai et al., "Adaptive Loop Filtering for Video Coding," *IEEE J. Selected Topics in Signal Processing*, vol. 7, no. 6, 2013, pp. 934–945.
5. J. Zhang, D. Zhao, and W. Gao, "Group-Based Sparse Representation for Image Restoration," *IEEE Trans. Image Processing*, vol. 23, no. 8, 2014, pp. 3336–3351.
6. P. List et al., "Adaptive Deblocking Filter," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, 2003, pp. 614–619.
7. X. Zhang et al., "Adaptive Loop Filter with Temporal Prediction," *Proc. Picture Coding Symposium* (PCS), 2012, pp. 437–440.
8. S. Wenger et al., "Adaptation Parameter Set (APS)," *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-F747*, 2011; http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11.
9. K. Dabov et al., "Image De-noising by Sparse 3D Transform-Domain Collaborative Filtering," *IEEE Trans. Image Processing*, vol. 16, no. 8, 2007, pp. 2080–2095.

Siwei Ma is an associate professor at the Institute of Digital Media, School of Electronic Engineering and Computer Science (EECS), Peking University, Beijing. His research interests include image and video coding, video processing, video streaming, and transmission. Ma received a PhD in computer science from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, and did postdoctorate work at the University of Southern California. Contact him at fswma@pku.edu.cn.

Xinfeng Zhang, the corresponding author for this article, is a research fellow at Nanyang Technological University, Singapore. His research interests include image and video processing, and image and video compression. Zhang received a PhD in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. Contact him at xfzhang@ntu.edu.sg.

Jian Zhang is a postdoctoral fellow at National Engineering Laboratory for Video Technology (NELVT), Peking University, Beijing. His research interests include image/video coding and processing, compressive sensing, sparse representation, and dictionary learning. Zhang received a PhD in computer science from the Harbin Institute of Technology, China. He was the recipient of the Best Paper Award at the 2011 IEEE Visual Communication and Image Processing. Contact him at jian.zhang@pku.edu.cn.

Chuanmin Jia is a doctoral student at the Institute of Digital Media, EECS, Peking University. His research interests include image processing and video compression. Jia received his BS in computer science from Beijing University of Posts and Telecommunications. Contact him at cmjia@pku.edu.cn.

Shiqi Wang is a postdoc fellow in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include video compression and image video quality assessment. Wang received a PhD in computer application technology from Peking University. Contact him at sqwang@pku.edu.cn.

Wen Gao is a professor of computer science at the Institute of Digital Media, EECS, Peking University. His research interests include image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interfaces, and bioinformatics. Gao received a PhD in electronics engineering from the University of Tokyo. Contact him at wgaog@pku.edu.cn.