Optimized Non-local In-Loop Filter for Video Coding

Xuewei Meng*, Chuanmin Jia*, Shanshe Wang*, Xiaozhen Zheng† and Siwei Ma*
*Institute of Digital Media, Peking University, Beijing, China
Email: {xwmeng, cmjia, sswang, swma}@pku.edu.cn
†SZ DJI Technologies Co., Ltd. Shenzhen, China
Email: xiaozhen.zheng@dji.com

***Abstract*—In order to compensate the shortcomings of existing in-loop filters only based on local correlation in video coding standards, many non-local based loop filters with high coding performance and computational complexity are proposed. In this paper, we propose a fast block matching algorithm, adaptive two-step block matching algorithm, based on our previous work, structure-driven adaptive non-local filter (SANF) which is computationally intensive because of the high complexity of block matching and singular value decomposition (SVD). Our proposed algorithm based on image spatial statistical characteristics utilizes fixed template to select adaptive number of similar blocks according to image content, which can reduce up to 75.2% search candidates compared to exhaustive search in SANF and the adaptive determination strategy can remove blocks with less relation to reference block in similar block group which have little help for compression performance, and the remove of them can reduce the computational complexity of SVD. Our proposed optimization algorithm can save encoding and decoding time significantly with negligible performance loss, which achieves 70.7%, 84.4%, 80.82% and 81.95% decoding time saving with only 0.13%, 0.05%, 0.13% and 0.15% increases of BD-rate for AI, RA, LDB and LDP configurations, respectively compared to original SANF in JEM-7.0.**

I. Introduction

The state-of-the-art video coding standard, High Efficiency Video Coding (HEVC), developed by the Joint Collaborative Team on Video Coding (JCT-VC) with ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG), has been released in Jan. 2013 [1]. HEVC can achieve about 50% bit-rate saving with equivalent perceptual quality compared to H.264/AVC. To further improve the compression efficiency, the two groups, i.e., VCEG and MPEG, are working together on the exploration activity in the Joint Video Exploration Team (JVET) [2]. The JVET was found in Oct. 2015, and released the reference software, Joint Exploration Model (JEM), which is based on the HEVC Model (HM) [3]. Until now, JEM-7.0 has achieved 19.85%, 28.51%, 22.33% and 25.99% bit rate saving for All Intra Main 10 (AI), Random Access Main 10 (RA), Low-Delay B Main 10 (LDB) and Low-Delay P Main 10 (LDP) configurations, respectively compared to HM-16.16 [4].

In the above video coding standards, in-loop filters play an important role in improving coding efficiency by reducing compression artifacts, such as blocking, ringing and blurring artifacts. There are two kinds of in-loop filters in HEVC, i.e., Deblocking Filter (DF) [5] and Sample Adaptive Offset (SAO) [6]. In JEM, besides DF and SAO, another two kinds of in-loop filters have been adopted, i.e., Bilateral Filter [7] and Adaptive In-Loop Filter (ALF) [8]. Bilateral Filter is the first in-loop filter in JEM located before DF, and ALF is the last one after SAO. However, these in-loop filters only focus on local correlation within image patches without fully considering image nonlocal self-similarities, which limits the filtering performance. To solve this problem, researchers have proposed many filtering methods based on image nonlocal similarities [9]–[16].

In [9], Buades *et al.* proposed the famous nonlocal means filter (NLM) to reduce compression artifacts by predicting each pixel with weighted average of nonlocal pixels. The weights are determined by the similarity between image patches located at the source and target coordinates. Considering the high efficiency of NLM in denoising problem, Matsumura *et al.* first introduced NLM into HEVC to compensate shortcomings of the existing in-loop filters only based on local correlation within image patches [10].

However, the pixel-independent filtering does not fully utilize the correlation within patches, some more complex algorithms based on image nonlocal similarities are proposed by collaboratively utilizing image similarity both insides patches and among similar patches. The well-known block-matching and 3D filtering (BM3D) [11] stacks nonlocal similar patches into 3D matrices and shrinks 3D transform coefficients of similar patches based on the image-sparse prior model to remove noise. In video coding area, Zhang *et al.* [12]–[14] proposed a nonlocal adaptive loop filter (NALF) utilizing image nonlocal prior knowledge, which improves the traditional low-rank based filters by adaptively estimating compression noise for every similar patch group. Our previous work, structure-driven adaptive non-local filter (SANF) [15] is a simplification version of NALF using global noise level for all similar patch groups. Although these methods significantly improve coding efficiency, they also bring in high computation burdens, e.g., the encoding time increase of NALF is up to 447% for AI and about 206% for LDB and RA configurations with 8 internal bit depth and the encoding time increase of SANF is 289%, 126%, 148% for AI, LDB and RA configurations with 8 internal bit depth, respectively. The decoding time increase of SANF is 10718%, 9219%, 7764% for the three configurations respectively against the anchor decoder. Recently, deep learning based in-loop filters are also investigated [16]. However, the complexity is even higher than SANF.

In this paper, in order to reduce the complexity of SANF without sacrificing its performance, we propose a fast block matching algorithm, adaptive two-step block matching algorithm, to search various number of similar blocks based on fixed template according to the spatial statistical characteristics of image, which can reduce the complexity of block matching process significantly compared with exhaustive search. In addition, the adaptive determination strategy for the number of similar blocks can remove less similar blocks which are of little benefit to filtering performance. The reduction of similar block number can directly decrease the number of iterations during singular value decomposition (SVD) and further reduce the complexity of filtering process. Our proposed optimized SANF achieves up to 70.7%, 84.4%, 80.82% and 81.95% decoding time saving with only 0.13%, 0.05%, 0.13% and 0.15% BD-rate increasing for AI, RA, LDB and LDP configurations, respectively compared to original SANF in JEM-7.0 [17].

The rest of the paper is organized as follows. Section II briefly reviews SANF algorithm in video coding. In Section III, the proposed adaptive two-step block matching algorithm and its implementation details are elaborated. Section IV shows experimental results of the proposed method. Finally, conclusions are drawn in section V.

II. Review of Structure-driven Adaptive Non-local Filter

SANF is proposed to enhance the quality of the reconstructed frames by simultaneously enforcing the intrinsic local sparsity and the nonlocal self-similarity of each frame in video. There are mainly three sub-modules, block matching and group construction by means of exhaustive search, group-based filtering by applying SVD to similar block groups and image reconstruction according to the filtered group after SVD. The details are given below.

A. Block Matching and Group Construction

As illustrated in Fig.1, the input reconstructed frame is firstly divided into K overlapped blocks of size $\sqrt{B_s} \times \sqrt{B_s}$, and each block is denoted by the vector $x_k \in \mathbb{R}^{B_s}, k = 1, 2, ..., K$. Then, for each block x_k, denoted by small red square in Fig.1, c best matched blocks are collected by means of exhaustive search within the corresponding search window (big blue square), which comprise the set S_{x_k}. Herein, sum of squared differences (SSD) is selected as the similarity criterion between search candidate block and reference block, and the c blocks with minimum SSD are selected as similar blocks. The similar block selection for each reference block utilizes exhaustive search manner according to the raster scan order in the given search window. After that, all the similar blocks in S_{x_k} are stacked into a matrix of size $B_s \times c$, denoted by X_{G_k}, where each column of the matrix corresponds to one block, i.e. $X_{G_k} = [x_{G_k \otimes 1}, x_{G_k \otimes 2}, ..., x_{G_k \otimes c}]$.

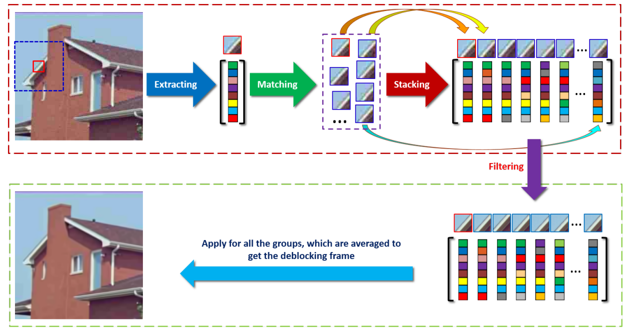


Fig. 1. Illustrations for Structure-driven adaptive non-local filter(SANF) [15]

B. Group-based Collaborative Filtering

For each group $X_{G_k} = [x_{G_k \otimes 1}, x_{G_k \otimes 2}, ..., x_{G_k \otimes c}], k = 1, 2, ..., K$, it is decomposed by SVD as follows,

$$X_{G_k} = U_{G_k} \Sigma_{G_k} V_{G_k}^T = \sum_{i=1}^{m} \gamma_{x_{G_k \otimes i}} (u_{G_k \otimes i} v_{G_k \otimes i}^T), \quad (1)$$

where $\gamma_{x_{G_k}} = [\gamma_{x_{G_k \otimes 1}}; \gamma_{x_{G_k \otimes 2}}; ...; \gamma_{x_{G_k \otimes m}}]$ is a column vector, $\Sigma_{G_k} = diag(\gamma_{x_{G_k}})$ is a diagonal matrix with the elements of $\gamma_{x_{G_k}}$ on its main diagonal, and $u_{G_k \otimes i}$, $v_{G_k \otimes i}$ are the columns of U_{G_k} and V_{G_k}, separately [15].

To suppress the compression noise, the hard thresholding operation is applied to the singular values, $\gamma_{x_{G_k}}$

$$\alpha_{G_k} = hard(\gamma_{x_{G_k}}, \tau), \quad (2)$$

where $hard(x, a) = x \odot 1(abs(x) - a)$ denotes the operator of hard thresholding and $\odot$ stands for the element-wise product of two vectors. τ denotes the threshold which is off-line trained and determined by QP and frame type.

C. Image Reconstruction

The filtered image can be reconstructed from the shrunk singular values α_{G_k}, and the reconstruction for group $\hat{X}_{G_k}$ can be formulated as,

$$\hat{X}_{G_k} = \sum_{i=1}^{m} \alpha_{G_k \otimes i} (u_{G_k \otimes i} v_{G_k \otimes i}^T), \quad (3)$$

Finally, the whole frame can be reconstructed by reallocating the blocks in all the reconstructed groups to its corresponding position in frame $\hat{X}$.

Moreover, SANF is located between DF and SAO with frame level on/off control. For each frame, there is only one flag signaled in frame header. At the encoder side, the frame level flag is determined according to the rate-distortion costs w/o SANF. If the frame level flag is on, the corresponding frame will be filtered at both encoder and decoder. Otherwise, SANF is switched off at both sides. This strategy can ensure the positive effect of SANF and avoid the useless filtering computation. However, the computational complexity of SANF is too expensive to apply to video coding standards. In order to solve this problem, we propose a fast block matching algorithm, adaptive two-step block matching algorithm.

III. Proposed Fast Block Matching Algorithm

A. Motivation

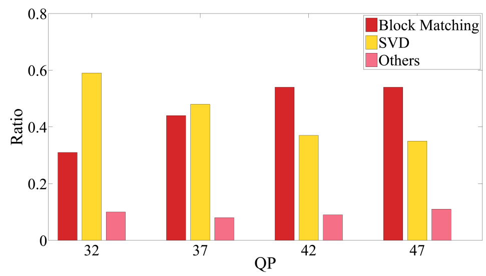
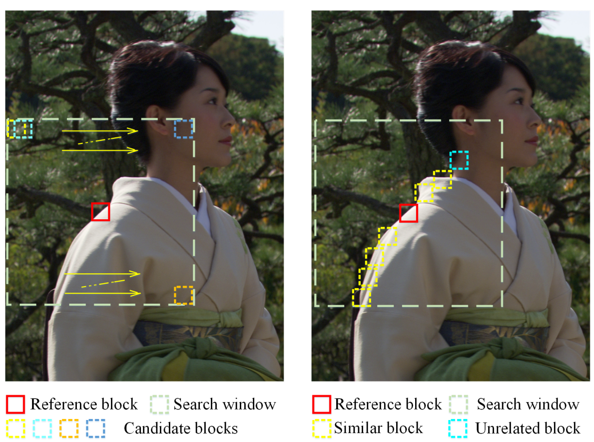


Fig. 2. Percentage of running time for main sub-modules in SANF



(a) Block matching process (b) Similar blocks selected

Fig. 3. Block matching algorithm of original SANF (exhaustive search)

In Fig.2, we provide the running time distribution of different sub-modules in SANF. The running time is obtained by filtering the test sequence *BQTerrance*, which is compressed by JEM-7.0 under RA configuration with quantization parameter (QP) 32, 37, 42, 47. We can see that block matching and SVD take up most of the running time of SANF. Specifically, block matching sub-module takes up to 54% running time of SANF when QP is 47. In this paper, we mainly focus on the block matching sub-module to reduce its computation complexity.

As shown in Fig.3 (a), in block matching sub-module, SANF uses exhaustive search to find c ($c = 30$) most similar blocks (including reference block itself) for reference block (small red square) in the search window (big green square) with size $W_s \times W_s (W_s = 33)$ pixel by pixel. The block matching process starts from the top-left block (small yellow square) and ends up with the bottom-right one in search window according to raster scan order (yellow arrow). For each candidate block in the search window, the SSD should be calculated between it and its reference block. Then, all the SSD values should be sorted in ascending order and the first c blocks with smallest SSD values are selected as similar ones.

Quantitatively, if we use the method of exhaustive search to find 30 nearest blocks from a 33×33 search window, the number of candidate blocks is 1089. On the one hand, the number of candidate blocks in search window is much more than that used in SANF, and most of them are not utilized in sequent filtering operation. Hence, there is much room for speeding up on block matching sub-module by reducing candidate blocks in search window. On the other hand, original SANF selects fixed c most similar blocks for all reference blocks without considering the diversity of image content, where there may not be c similar blocks for some specific reference blocks. In other words, if we choose c most similar blocks according to SSD, some blocks may not be related to the reference block as shown in Fig.3 (b). The blocks labelled by small blue square have little relation to the reference block. These blocks make little contributions for collaborative filtering. Therefore, we can further reduce computational complexity of block matching sub-module by removing these blocks with less relation and decreasing the size of similar block groups, which can directly reduce computational complexity of SVD.

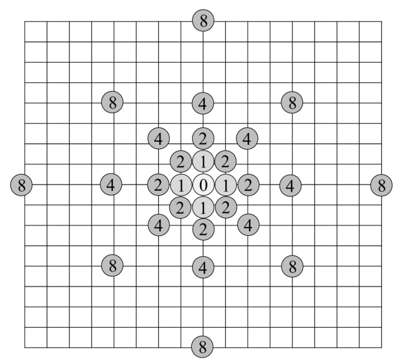


Fig. 4. Block matching template

B. Proposed Fast Block Matching Algorithm

In order to reduce the number of search candidates and remove the blocks with less relation in similar block group, we propose a fast block matching algorithm to search adaptive number of similar blocks by fixed template according to spatial statistical characteristics of images. Details are given below,

1) Step one. As shown in Fig.4, an eight-point diamond template is utilized to search most similar blocks of the reference block in search window with size $W_s \times W_s$ according to the characteristic that similar blocks usually gather in several areas. The center point of eight-point diamond search is the reference block denoted by point with label "0" in Fig.5. Select at most $T (T = 5)$ most similar blocks (golden points in Fig.5) from all the blocks searched during the process with SSD smaller than the threshold ϵ and record their coordinates. Thus

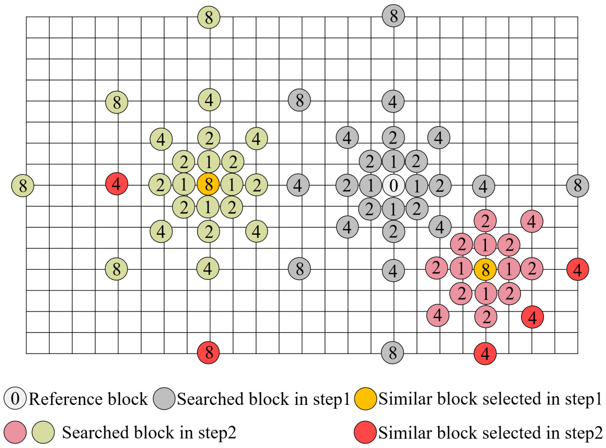


Fig. 5. Adaptive two-step block matching process

the number of blocks selected in step one may less than T. We use $m(m < T)$ to represent the number of blocks selected in the end.

2) Step two. As shown in Fig.5, we take the m selected similar blocks denoted by golden point in step one as center point, respectively, to search similar blocks by using eight-point diamond template. The blocks selected by step two are denoted by red points in Fig.5. The number of selected blocks in the end is no more than c and may be different from each other.

C. Parameter ϵ estimation

According to the above description, the setting of ϵ is of great significance in our proposed algorithm. This subsection presents the details to give an adaptive and robust estimation of ϵ according to the characteristics of video.

The similarity criterion between different blocks is SSD. Thus, the parameter ϵ is related to the size of reference block and the bit depth of video. Hence, we propose to estimate the parameter ϵ from block size and bit depth.

$$\epsilon = B_s \times (1 << (2 \times bitDepth)) \times \lambda, \tag{4}$$

where $B_s(B_s = 36)$ is pixel number of each patch, $bitDepth$ is the bit depth of test sequence, which is usually 8 or 10. $\lambda(\lambda = 0.06)$ represents the parameter we obtain according to the statistics of SSD between reference block and selected similar block by exhaustive search.

TABLE I
NUMBER OF SEARCH CANDIDATES

Sequence	Proposed	SANF	Ratio
MarketPlace	273	1072	74.6%
BQTerrace	319	1072	70.3%
BasketballDrive	306	1072	71.5%
RitualDance	266	1072	75.2%
Cactus	291	1072	72.9%
Average	291	1072	72.9%

The search candidate number of the proposed fast block matching algorithm and exhaustive search in SANF is shown in TABLE I. We use five sequences in Class B as test videos, and test Y component of the first frame in each sequence under AI configuration when QP is 47. The number of search candidates in TABLE I is the average number of every reference blocks' candidates in one frame. The size of search window is 33×33. The search candidate number of exhaustive search is 1072 rather than 1089 because there is no padding if search window cross image border. It can be seen that our proposed method can eliminate 72.9% less similar search candidates on average, which can save encoding and decoding time significantly. And the remove of less similar blocks can make the size of group used for SVD smaller than original SANF, hence the computation complexity of SVD can be further reduced.

IV. Experimental Results

In this section, we test the performance of original SANF and the proposed fast block matching algorithm by implementing them into JEM-7.0. Four configurations used in JVET were tested: AI, RA, LDB and LDP. Four typical QP values are 32, 37, 42 and 47. The video sequences in "Joint Call for Proposals on Video Compression with Capability beyond HEVC" [18] are utilized as test sequences in our experiment. The first two seconds of these sequences are encoded for performance evaluation. And coding performance is measured by Bjontegaard's method [19] in terms of BD-rate (Y component). 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 2013, 64bit. The tests run on Windows 10 operating system with 64 bit and CPU in the test is Intel(R) Core i7-8700 @ 3.20GHz.

We integrate SANF into JEM-7.0 and apply LCU level on/off control to it. We also retrain the parameter τ under configurations in JEM-7.0. Due to the introduction of new coding tools and higher accuracy of prediction module, the coding gains of SANF in JEM-7.0 is not as high as HM-12.0. TABLE II summarizes the results of SANF and proposed method. In the two tests, anchor is generated by JEM-7.0 with default configuration. EncT_JEM7.0/DecT_JEM7.0, EncT_SANF/DecT_SANF, EncT_Pro/DecT_Pro represent the encoding/decoding time of JEM-7.0, SANF and proposed method respectively. The decoding time of original SANF is up to 7171.03%, 3103.39%, 3817.04% and 5298.28% for AI, RA, LDB and LDP respectively, compared with JEM-7.0 decoder. The decoding time of our proposed optimized SANF is 2101.04% for AI, 484.18% for RA, 732.29% for LDB and 956.56% for LDP. Our method achieves 70.70%, 84.40%, 80.82% and 81.95% decoding time saving. Compared with JEM-7.0 encoder, the encoding time of the original SANF is 112.9% for AI, 107.8% for RA, 106.8% for LDB and 110.1% for LDP. The encoding time of our proposed optimized SANF is 104.2%, 101.7%, 101.2% and 102.7%. As shown in TABLE II, the average bit rate saving of Y component is 1.15%, 0.68%, 1.46% and 1.78% for AI, RA, LDB and LDP,

TABLE II
EXPERIMENTAL RESULTS OF SANF AND PROPOSED OPTIMIZED SANF, ANCHOR: JEM-7.0

		Original SANF				Proposed Optimized SANF			
Sequence	Resolution	AI	RA	LDB	LDP	AI	RA	LDB	LDP
MarketPlace	1920x1080	-0.61%	0.15%	-1.00%	-0.69%	-0.44%	0.34%	-0.76%	-0.48%
BQTerrace	1920x1080	-2.14%	-2.63%	-3.43%	-5.09%	-1.90%	-2.14%	-2.74%	-4.23%
BasketballDrive	1920x1080	-1.14%	-1.11%	-1.98%	-2.61%	-1.11%	-1.25%	-2.11%	-3.00%
RitualDance	1920x1080	-1.19%	-1.09%	-1.69%	-1.91%	-1.00%	-1.38%	-1.77%	-1.92%
Cactus	1920x1080	-1.35%	-1.29%	-1.75%	-1.93%	-1.15%	-1.12%	-1.73%	-1.73%
Campfire	3840x2160	-1.11%	-0.91%	-1.03%	-1.23%	-1.06%	-1.17%	-1.12%	-1.26%
CatRobot1	3840x2160	-1.28%	-0.20%	-1.42%	-1.71%	-1.09%	0.05%	-1.04%	-1.16%
DaylightRoad2	3840x2160	-1.17%	-0.23%	-1.58%	-1.67%	-1.04%	0.12%	-1.27%	-1.18%
FoodMarket4	3840x2160	-0.74%	1.23%	0.41%	0.20%	-0.75%	1.05%	0.09%	-0.26%
ParkRunning3	3840x2160	-0.77%	-0.74%	-1.09%	-1.19%	-0.71%	-0.77%	-0.90%	-1.06%
Average		-1.15%	-0.68%	-1.46%	-1.78%	-1.02%	-0.63%	-1.33%	-1.63%
EncT_SANF / EncT_JEM7.0 (%)		112.9%	107.8%	106.8%	110.1%	104.2%	101.7%	101.2%	102.7%
DecT_SANF / DecT_JEM7.0 (%)		7171.03%	3103.39%	3817.04%	5298.28%	2101.04%	484.18%	732.29%	956.56%
(EncT_SANF - EncT_Pro) / EncT_SANF (%)		-	-	-	-	7.71%	5.66%	5.24%	6.72%
(DecT_SANF - DecT_Pro) / DecT_SANF (%)		-	-	-	-	70.70%	84.40%	80.82%	81.95%

respectively. The average coding gains of proposed method are 1.02%, 0.63%, 1.33% and 1.63% for AI, RA, LDB and LDP, respectively. Note that the proposed method can achieve significantly time saving with negligible performance loss.

V. CONCLUSION

In this paper, an optimized non-local block matching algorithm is proposed to reduce the computational complexity of SANF. In the proposed method, block matching process of SANF is optimized by a template based fast search algorithm according to image spatial statistical characteristics with adaptive number of similar blocks. Experimental results demonstrate that our proposed algorithm can save 79.4% overall decoding time on average with negligible performance loss, which improves the practicability of SANF for future video coding standards.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (61632001), National Basic Research Program of China (973 Program, 2015CB351800)and the Top-Notch Young Talents Program of China, which are gratefully acknowledged.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] "Proposed revised requirements for a future video coding standard," *MPEG Doc.M36183, Warsaw, Poland*, 2015.

[3] F. Bossen, D. Flynn, and K. Sühring, "HEVC reference software manual," *JCTVC-D404, Daegu, Korea*, 2011.

[4] G. J. Sullivan and J.-R. Ohm, "Meeting report of the 8th JVET meeting," *JVET of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC29/WG11, Doc. JVET-H1000, 8th Meeting, Macao, CN*, 2017.

[5] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614–619, 2003.

[6] C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, S.-M. Lei, J.-H. Park, and W.-J. Han, "Sample Adaptive Offset in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1755–1764, 2012.

[7] J. Ström, P. Wennersten, K. Andersson, and J. Enhorn, "Bilateral filter strength based on prediction mode," *JVET of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC29/WG11, Doc. JVET-E0032, 5th Meeting, Geneva, CH*, 2017.

[8] M. Karczewicz, L. Zhang, W.-J. Chien, and X. Li, "Geometry transformation-based adaptive in-loop filter," in *Picture Coding Symposium (PCS)*. IEEE, 2016, pp. 1–5.

[9] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2005, pp. 60–65.

[10] M. Matsumura, Y. Bandoh, S. Takamura, and H. Jozawa, "Inloop filter based on non-local means filter," *JCTVC-E206, Geneva, CH*, 2011.

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

[12] X. Zhang, R. Xiong, W. Lin, J. Zhang, S. Wang, S. Ma, and W. Gao, "Low-rank based nonlocal adaptive loop filter for High Efficiency Video Compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2177–2188, 2017.

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

[14] S. Ma, X. Zhang, J. Zhang, C. Jia, S. Wang, and W. Gao, "Nonlocal in-loop filter: The way toward next-generation video coding?" *IEEE MultiMedia*, vol. 23, no. 2, pp. 16–26, 2016.

[15] J. Zhang, C. Jia, N. Zhang, S. Ma, and W. Gao, "Structure-driven adaptive non-local filter for High Efficiency Video Coding (HEVC)," in *Data Compression Conference (DCC)*. IEEE, 2016, pp. 91–100.

[16] C. Jia, S. Wang, X. Zhang, S. Wang, and S. Ma, "Spatial-temporal residue network based in-loop filter for video coding," *arXiv preprint arXiv:1709.08462*, 2017.

[17] J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 7 (JEM7)," *JVET of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC29/WG11, Doc. JVET-G1001, 7th Meeting, Torino, IT*, 2017.

[18] A. Segall, V. Baroncini, J. Boyce, J. Chen, and T. Suzuki, "Joint call for proposals on video compression with capability beyond HEVC," *JVET of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC29/WG11, Doc. JVET-H1002, 8th Meeting, Macao, CN*, 2017.

[19] G. Bjontegarrd, "Calculation of average PSNR differences between RD-curves," *VCEG-M33, 13th VCEG Meeting, Austin, TX, USA*, 2001.