Optimized in-loop filtering in versatile video coding using improved fast guided filter

Lakshmi Amrutha Valli Pamidi, Purnachand Nalluri
School of Electronics Engineering, VIT-AP University, Inavolu, India

Article Info

Article history:

Received Oct 19, 2023
Revised Nov 30, 2023
Accepted Dec 16, 2023

Keywords:

Deblocking filter
Improved fast guided filter
In-loop filtering
Versatile video coding
VVC test model

ABSTRACT

Devices with varying display capabilities from a common source may face degradation in video quality because of the limitation in transmission bandwidth and storage. The solution to overcome this challenge is to enrich the video quality. For the mentioned purpose, this paper introduces an improved fast guided filter (IFGF) for the contemporary video coding standard H.266/VVC (versatile video coding), a continuation of H.265/HEVC (high efficiency video coding). VVC includes several types of coding techniques to enhance video coding efficiency over existing video coding standards. Despite that, blocking artifacts are still present in the images. Hence, the proposed method focuses on denoising the image and the increase of video quality, which is measured in terms of peak signal-to-noise (PSNR). The objective is achieved by using an IFGF for in-loop filtering in VVC to denoise the reconstructed images. VTM (VVC test model)-17.2 is used to simulate the various video sequences with the proposed filter. This method achieves a 0.67% Bjontegaard delta (BD)-rate reduction in low-delay configuration accompanied by an encoder run time increase of 4%.

Corresponding Author:

Purnachand Nalluri
School of Electronics Engineering, VIT-AP University
Inavolu, Andhra Pradesh, India
Email: chanduinece@gmail.com

1. INTRODUCTION

Versatile video coding (VVC) is the current video coding standard that came into existence through the joint video experts team (JVET), a collaboration between the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) video coding experts group (VCEG) and the ISO/IEC moving picture experts group (MPEG) [1]. VVC is the descendant of high efficiency video coding (HEVC) and it promises to provide significant progress in compression efficiency over its predecessors [2]. HEVC is followed by the advanced video coding (AVC) standard [3]. VVC is expected to be utilized in a large range of applications, involving high-resolution video broadcasting, virtual reality, and 360-degree video.

VVC achieves its improved compression efficiency by employing advanced video coding techniques such as block splitting, prediction, and transform coding. It also introduces new tools such as enhanced motion vector prediction, and compensation followed by an effective entropy coding scheme [1], [4]. The adoption of new tools in VVC and also with the new coding structures namely prediction unit (PU), coding unit (CU), and transform unit (TU) deployed in video coding standards to split the frames for increasing efficient compression causes discontinuities which are appeared along the edges of the blocks. These discontinuities are stated as artifacts [1], [4]. In-loop filters belonging to VVC standard, are employed for reducing artifacts and improving the visual quality of the particular compressed video. The blocking artifacts are categorized as blocking, ringing, and blurring artifacts. Filters are arranged in the decoder either in-loop or out-loop. However, these filters can also introduce complexity in the encoding and decoding processes, which creates an issue with real-time video applications or gadgets with limited processing power [5].

VVC standard is included with five in-loop filters as shown in Figure 1 and also with several updated coding tools to enhance the efficiency of compression over the other previous standards [1], [3]. These filters have been referred as in-loop because they are present in the encoding loop where an iterative process is performed on the video frames before storing them in the picture buffer. These filters strengthen the quality of the video frames, unlike the post-processing algorithms or filters. The deblocking filter (DBF) is responsible for clearing blocking artifacts present in the decoded frames. Sample adaptive offset (SAO) filter is used for removing ringing artifacts and adaptive loop filter (ALF) is applied to further optimize the quality of the frames obtained from the DBF. Cross-component adaptive loop filter (CC-ALF) is specifically used to correct the chroma samples by exploiting the correlation between the luma and chroma samples [6]–[10].

As mentioned earlier, discontinuities are increased due to the size of the blocks varying up to 128×128 in VVC. Hence a control mechanism is established in the algorithm of the deblocking filter to overcome these artifacts. The algorithm depends on key factors like boundary strength, filter length, and separate filters for both luma and chroma samples. DBF is used at regions where artifacts are more, to obtain smoothness across the edges. There is no need for the filter at natural edges having low quantization parameter (QP) value [7].

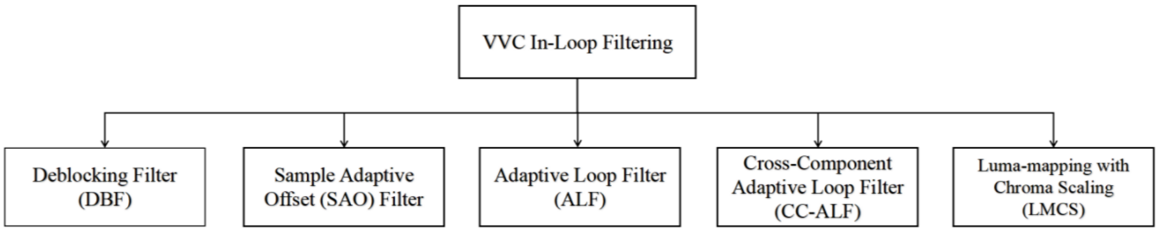

Figure 1. In-loop filters in VVC

The algorithm of DBF in VVC is related to the DBF in HEVC and AVC coding standards [10]–[12]. The primary step is to calculate the boundary strength (BS) along the block boundaries. The BS value calculation and the filtering operations rely on the pixel values. There are two offset or threshold parameters. If the pixel difference values are under these parameters, then the filtering operations are performed for the corresponding samples. The threshold parameter value increases with the increase in QP value as they are reliant on QP. The threshold values are also dependent on the type of coding [7]. The proposed block diagram of the modified VVC encoder with improved fast guided filter (IFGF) inserted between DBF and SAO filter is shown in Figure 2. The main objective of the three filters (DBF, IFGF, and SAO) is to reduce blocking artifacts, coarsening of edges, and ringing artifacts.

The need for ultra-high-definition (UHD) videos increases rapidly in contrast to high-definition (HD) videos. The quality of the videos changes concerning the type of gadget, transmission bandwidth, and storage space. The quality of videos degrades in certain circumstances even with the presence of in-loop filters. The quality and the compression efficiency are dependent on the spatial and temporal prediction. These two redundancies are affected by the addition of noise in various environments [13]. Many coding tools are introduced to perform better redundancy techniques. Still, the bitrate of the videos increases, and the quality of the videos decreases with the addition of noise. The average bitrate is increased by a huge percentage with the presence of noise in images [14]. Hence, denoising techniques [15], [16] are required to combine with in-loop filters of coding standards for enhancing coding efficiency. Therefore, the proposed method of this paper is an improved fast-guided filter next to the deblocking filter in the VVC. The main contributions to overcome the problem stated are: i) an IFGF is proposed for its computational efficiency, exceptional edge preservation, simplified parameter tuning, minimal artifacts, and noise reduction; ii) to modify the fast guided filter (FGF) in the form of improved FGF; and iii) applying IFGF for VVC to increase the quality of the videos. Up to our knowledge, the proposed technique is the first to use IFGF for in-loop filtering in VVC to improve video quality measured in peak signal-to-noise ratio (PSNR) (dB).

The paper discusses the following: a literature review describing various authors' work including methodology, results, and limitations in section 2, a suggested method using IFGF in VVC in-loop architecture in section 3, results obtained for various test sequences along with an explanation in section 4 and the summary of the work in section 5 ends the paper.

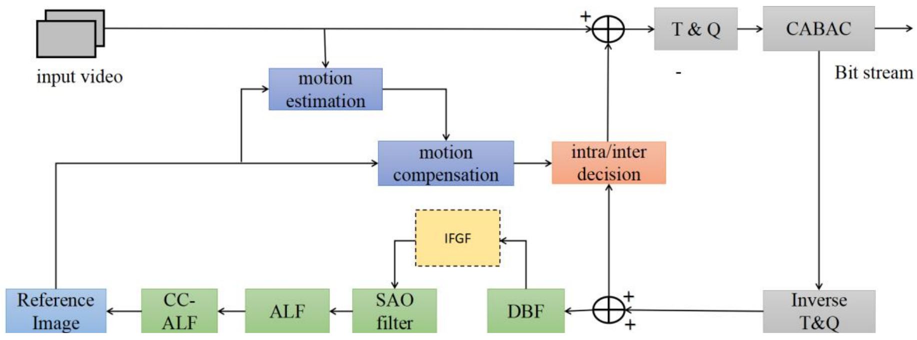

Figure 2. Block diagram of VVC encoder with In-loop filters and proposed filter

2. LITERATURE REVIEW

Much research has been done on these denoising techniques to boost compression efficiency and image quality. In this section, some of them are introduced. Wennersten *et al.* [17], a bilateral filter is suggested as an optimizing tool in HEVC. To avoid the complexity and to enhance the compression efficiency, implementation is done using a look-up table (LUT), where it is having 2202 bytes in size. This method is tested on HM version 5.0.1 and reports a 0.5% BD-rate reduction followed by an increase of 3% encoder time. The performance of bilateral filters on various video sequences is not discussed in terms of video quality and bit rate. Strom *et al.* [18], a bilateral filter is used for VVC. The filter is used in combination with the SAO filter. The input samples are applied to both filters which are in parallel, the output is obtained after clipping both samples from the filters. The method is tested on VTM 5.0 with a BD-rate reduction of 0.4% in all intra and 0.5% in random access configurations with an increase in the encoder and decoder run time. A limitation is present due to its position in the critical path of hardware implementations, which might make it difficult to process high resolutions at an appropriate frame rate.

Chen *et al.* [19], an adaptive guided filter is proposed for sharpening and smoothing the images. Filter parameters are assigned to each group's pixel depending on rate-distortion (RD) optimization, where groups are divided based on QP value. Simulation results for this filter achieve a 0.7% BD rate reduction with a 15% raise in encoder time and a 70% raise in decoder time on average. The specific strength of the proposed filter is not mentioned other than the increase in the quality of the videos. Yin *et al.* [20], an adaptive self-guided loop filter (ASGF) is mentioned to work along with a loop filter in video coding. The main thought is to identify strong edges and then accordingly the filter strength is adjusted by using a controlling parameter. On VTM 11.0, the suggested filter produced 0.41% and 0.46% BD-rate reductions in Random access and all intra-configuration, respectively. It has some restrictions in terms of the components to which it can be applied and the requirements that must be met for its implementation. Ikonin *et al.* [21], a filter in the Hadamard transform domain is proposed. The filter is applied directly to the decoded samples at the block level during reconstruction. The filter settings are chosen based on the QP. The filter operations depend on LUT to avoid multiplications and divisions, where the filter is compatible with hardware. The proposed method achieves a result of 0.5% BD-rate reduction followed by a 4% and 1% encoding and decoding time increase on VTM 2.0. It may not function effectively for images with complex textures or irregular patterns because the transform may not capture all the necessary features or structures, resulting in unfulfilling denoising or filtering. As a result of previous works, we propose a smoothing filter named improved fast guided filter based on guided image filtering to improve the in-loop filtering of VVC.

2.1. Review of guided filter

The guided filter has been used to smooth the given input image X in the guidance of image I for obtaining an output image Y. The guided filter is also controlled by filtering parameters comparable to filters in VVC. Based on the rate-distortion optimization, the parameters for the previous encoding frames are determined and used for the current frames [22]. A linear model driving the guided filter is;

$$Y_i = u_k I_i + v_k, \ \forall i \in w_k \tag{1}$$

where i and k denote the pixel index and local window index respectively. The radius r of the window (w) and u_k, v_k are applied in this case chosen to minimize errors in the filtering image.

$$u_k = \frac{\frac{1}{|W|}\sum_{i \in w_k} I_i X_i - \mu_k \bar{X}_k}{\sigma_k^2 + \epsilon} \tag{2}$$

$$v_k = \overline{X}_k - u_k \mu_k \tag{3}$$

Where μ_k stands for the average mean and σ_k stands for the variance of I respectively. ϵ denotes the controlling parameter for the degree of smoothness.

$$\mu_k = \frac{1}{R^2} \sum_{i \in w_k} I_i \tag{4}$$

$$\sigma_k^2 = \frac{1}{R^2} \sum_{i \in w_k} I_i^2 - \mu_k^2 \tag{5}$$

The pixel values may belong to several pixel windows whose coefficients u_k and v_k are associated with them. Hence the final average filtered output Y is given by;

$$Y_i = \bar{u}_i I_i + \bar{v}_i \tag{6}$$

$$\overline{X}_k = \frac{1}{R^2} \sum_{i \in w_k} X_i \tag{7}$$

where, $\overline{X}_k$ stands for the average value of the X (input image) within the selected window w_k.

$$\bar{u}_i = \frac{1}{R^2} \sum_{k \in w_i} u_k \tag{8}$$

$$\bar{v}_i = \frac{1}{R^2} \sum_{k \in w_i} v_k c \tag{9}$$

$\bar{u}_i$ and $\bar{v}_i$ are the mean values of u and v in (8) and (9).

3. METHOD

The details of the mentioned filter IFGF are explained in detail in this section. The main benefits of a guided filter over a bilateral filter are speed, fine sharpening, and smoothing of edges, followed by an increase in complexity while implementing filters with large kernels. There are some challenges in regular guided filters. Integral sums are part of calculations involving bit depth and memory resources. A stripe-based method is used for optimizing integral sums. FGF is proposed in the paper [23], [24], where box filters are being used as prime blocks for filtering to face the above-mentioned challenges of the guided filter. These box filters act as low-pass filters whose output is a downsampled image. By this motivation, an improved FGF is proposed to overcome the additional errors caused by FGF. An additional prefiltering step is included in the form of a box filter again to downsample the image.

When an input image is immediately subsampled, aliasing may occur in the output image, resulting in an extra error. To address this, FGF employs a prefiltering step in addition to downsampling in the form of a simple box filter (low pass filter). By lowering the spectral components, aliasing can be avoided or at least reduced by employing the use of a low-pass filter before decimation. A basic SxS box filter, where S is the down-sampling ratio, can be used to perform pre-filtering. Although it appears that the output of SxS pre-filters needs to be calculated for the entire resolution of the input image, it may be further improved because it is followed immediately by a decimation step.

The box filter and size SxS decimation are merged into a single stage known as the decimated box filter. There is no need to determine the whole outputs of the filtering step considering only a small fraction of the filtering stage's output pixels would propagate further after decimation. Instead, the input image is divided into SxS independent blocks, and the mean value of every block is transmitted to the next stage using a decimated box filter. Because each input pixel is only used for a single output pixel, the complexity of a decimated box filter is O(N) irrespective of filter size. This eliminates the need for integral picture calculation. Furthermore, the decimated box filter can be generated instantaneously while reading samples of input from memory, no additional memory is required. The pre-filtering step involved in the above IFGF is the main step that deviates from the fast-guided filter.

$$I' = \left(\frac{1}{s^2} \sum_{i \epsilon w_k''} I_i\right) = \downarrow_{SXS} = \mathrm{DBF}(I) \tag{10}$$

$$I^{2\prime} = (\frac{1}{s^2} \sum_{i \epsilon w_k''} I_i^2 = \downarrow_{SXS} = \mathrm{DBF}(I^2) \tag{11}$$

Where apostrophe represents that the I is computed in the downsampled domain by a factor S. w_k'' is a window of size SxS and DBF is called a decimated box filter. The remaining calculations are similar to FGF as mentioned above.

$$\mu_k' = \frac{1}{k^2}\sum_{i \epsilon w_k'} I_i' \quad (12)$$

$$\sigma_k^2 = \frac{1}{k^2}\left(\sum_{i \epsilon w_k'} I_i^{2'}\right) - \mu_k'^2 \quad (13)$$

Where μ_k stands for the average mean and σ_k stands for the variance of I respectively. ϵ denotes the controlling parameter for the degree of smoothness.

$$\bar{u}_i = \left(\frac{1}{K^2}\sum_{k \epsilon w_k'} u_k'\right)\uparrow_{SXS}, \text{ where } u_k' = \frac{\sigma_k'^2}{\sigma_k'^2 + \epsilon} \quad (14)$$

$$\bar{v}_i = \left(\frac{1}{K^2}\sum_{k \epsilon w_k'} v_k'\right)\uparrow_{SXS}, \text{where } v_k' = (1-u_k')\mu_k' \quad (15)$$

Where, $\bar{u_i}$ and $\bar{v}_i$ are the average values of the filter coefficients.

Figure 3 represents the workflow in the proposed filter. Multiple frames are available for each video sequence in video coding. To execute guided filtering, the current frame is chosen as the guidance image, and the surrounding frames are handled as the input frames to be filtered. Each pixel in the current frame has a local window defined around it. This window is used to collect data from both the guide image and the frames around it. Usually, characteristics such as the filtering radius determine the size of this window. The mean and covariance of pixel values are determined individually for the guided image and every adjacent frame within the local window. This data characterizes the image's local statistics. Filtering coefficients are calculated using the local means and covariances. These coefficients specify how much influence each neighboring frame should have on the current pixel's filtering.

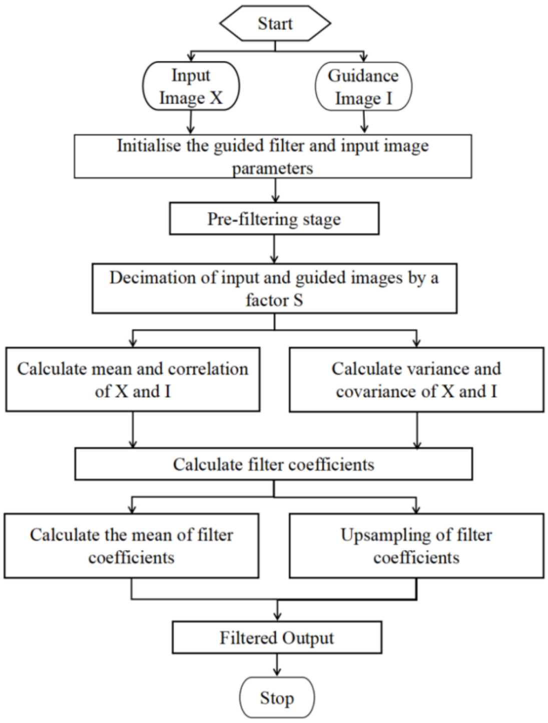

Figure 3. Flowchart of the proposed filter IFGF

In general, the coefficients are determined using linear regression or a similar method. The obtained coefficients can be used to perform weighted averaging of pixel values in adjacent frames. The current frame's filtered pixel value is obtained using weighted averaging. Each pixel in the current frame within the local window is subjected to the filtering procedure. The above steps are repeated for the entire video sequence. The guidance image along with the neighboring frames is changed based on the optimization parameters for every frame. The filtered frames are used in the motion estimation and prediction of the remaining frames. As a result, compression artifacts are reduced and overall video quality is improved.

4. RESULTS

The VTM tool version 17.2 [25], was used in evaluating the efficiency of the proposed technique as the reference software for the analysis of different video sequences. The list of five categories of real video sequences [26]–[28] is given in Table 1. Implementation was performed on an Intel Core i7 with 16GB of RAM. Five different classes of video sequences are chosen and 10 frames from each video sequence are used to test. The quantization parameter values of 22, 27, 32, 37 are chosen. The test was conducted in a low-delay configuration. Several criteria, including PSNR, encoding time, and bitrate, have been analyzed to completely evaluate the performance of the updated VTM. The study contained a diverse set of video sequences with varying frame rates, allowing for a full investigation of the VTM's performance across a wide variety of motion characteristics. This can help in the discovery of limitations that can be changed to optimise the encoding process. Analyzing performance factors such as PSNR, bitrate, and encoding time yielded important trade-offs required to obtain increased compression efficiency at the expense of acceptable video quality. It is crucial to make clear that the outcomes of these analyses are particular to the proposed filter and video sequences chosen. The proposed algorithm's performance is shown in the table. From Table 1, it is evident that the average PSNR is altered considerably, while the encoding time is significantly increased in comparison with the original algorithm. The coding effectiveness of the proposed filter in VVC is shown in Table 1.

Table 1. Simulation results for various video sequences in low-low-delay configuration

Class	Sequence	Qp=22			Qp=27			Qp=32			Qp=37		
		BD-PSNR	BD-Bitrate	ΔT (%)	BD-PSNR	BD-Bitrate	ΔT (%)	BD-PSNR	BD-Bitrate	ΔT (%)	BD-PSNR	BD-Bitrate	ΔT (%)
Class A	Honey Bee	+1.32	-0.45	8%	+1.21	-0.62	7%	+1.18	-0.69	6%	+1.09	-0.87	5%
3840x2160_120	Jockey	+1.23	-0.53	9%	+1.19	-0.64	7.5%	+1.11	-0.67	6.3%	+1.02	-0.78	5.5%
	Average	+1.27	-0.49	8.5%	+1.2	-0.63	7.2%	+1.14	-0.68	6.1%	+1.05	-0.82	5.2%
Class	Kimono	+1.19	-0.76	8%	+1.11	-0.91	6%	+1.06	-0.93	5%	+1.01	-1.16	4.5%
B	Park Scene	+1.11	+0.03	7%	+1.09	+0.05	6.3%	+1.04	+0.06	5%	+1.02	+0.19	4%
1920x1	Cactus	+1.09	-0.28	7%	+1.05	-0.45	6%	+1.03	-0.59	4%	+0.99	-0.96	4%
080_24	BQ Terrace	+1.08	-0.41	7.5%	+1.12	-0.54	5.5%	+1.06	-0.56	4%	+0.98	-0.65	3.5%
	Average	+1.11	-0.37	7.2%	+1.09	-0.47	6%	+1.04	-0.52	4.5%	+1.00	-0.69	4%
Class	Race Horses	+1.06	-0.58	6%	+1.01	-0.61	5%	+0.98	-0.63	3%	+0.96	-0.73	2%
C	BQ Mall	+1.02	-0.27	6%	+0.99	-0.34	5%	+0.99	-0.54	4%	+0.94	-0.59	2.5%
832x48	Party Scene	+1.05	-0.43	5%	+1.01	-0.52	4%	+0.99	-0.59	3%	+0.93	-0.67	3%
0_50	Basketball Drill	+1.06	-0.38	5%	+0.99	-0.47	4%	+0.96	-0.52	3%	+0.91	-0.63	2%
	Average	+1.04	-0.41	5.5%	+1.00	-0.48	4.5%	+0.98	-0.57	3.2%	+0.93	-0.65	2.3%
Class	Race Horses	+1.01	-0.56	5%	+0.99	-0.59	3%	+0.91	-0.67	2%	+0.89	-0.85	1%
D	BQ Square	+1.02	-0.49	4%	+1.01	-0.57	3%	+0.92	-0.69	2%	+0.89	-0.76	1.5%
416x24	Blowing Bubbles	+1.02	-0.43	3%	+0.98	-0.55	3%	+0.91	-0.52	2%	+0.87	-0.71	2%
0_50	Basketball Pass	+1.04	-0.38	4%	+1.1	-0.5	2%	+0.96	-0.59	2%	+0.89	-0.78	1%
	Average	+1.02	-0.46	4%	+1.02	-0.55	2.7%	+0.92	-0.61	2%	+0.88	-0.77	1.3%
Class E	FourPeople	+1.04	-0.52	4%	+0.96	-0.63	4.5%	+0.91	-0.69	3%	+0.84	-0.95	2.5%
1280x7	Johnny	+1.06	-0.41	5%	+0.99	-0.55	4%	+0.94	-0.67	3%	+0.85	-0.81	2%
20_60	Kristn and Sar	+1.01	-0.57	4.5%	+0.97	-0.69	4%	+0.91	-0.71	3.5%	+0.86	-0.96	2.5%
	Average	+1.03	-0.5	4.3%	+0.97	-0.62	4.1%	+0.92	-0.69	3.1%	+0.85	-0.90	2.3%
Average of all test sequences		+1.06	-0.44	5.7%	+1.04	-0.55	4.6%	+0.99	-0.61	3.7%	+0.94	-0.76	2.8%

From the statistics, the Kimono video sequence achieved a 1.19% performance gain with a QP value of 22 over the VTM-17.2 original version. The algorithm achieves the best results in low-delay configuration compared to the random access and all intra configuration. The performance of the video sequences in case of video quality also decreases with the increase in the QP value in all three configurations.

Figure 4 explains the rate-distortion of various video sequences where the suggested approach improves the original approach in terms of PSNR. RD curves of various video sequences of four different classes are shown in Figure 4. From the graphs, it is clear that the proposed method exhibits better performance in comparison with the original method. The curve nearer to the Y-axis exhibits better performance than the other.

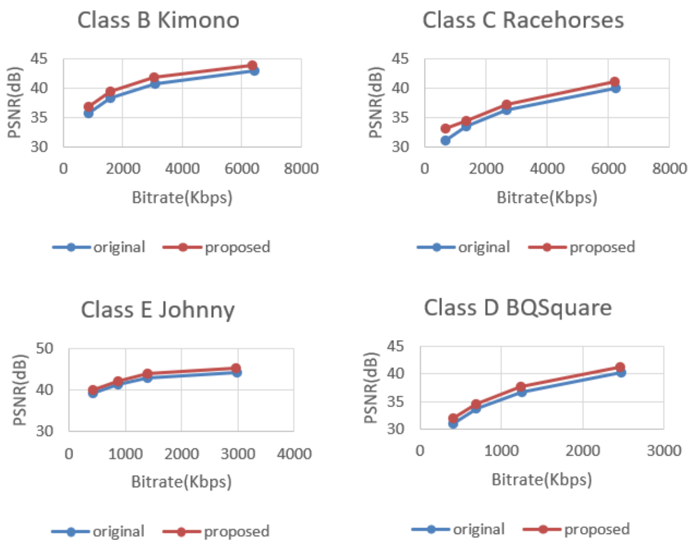

Figure 4. RD curves of various video sequences in low delay configuration

4.1. Discussion

The method recommended in this paper proposes a smoothing filter based on the guided image filter, which is placed in the VVC's in-loop filter architecture between the deblocking filter and the SAO. This approach is designed to reduce blocking artifacts, sharpen or restore image edges, and remove ringing artifacts. The IFGF is a prominent edge-aware filter that is used in a variety of image processing applications in contrast with other filters like median, gaussian, and bilateral. The quality of the approximation assessed in PSNR can be enhanced by implementing a pre-filtering stage before decimation. The proposed method of pre-filtering with decimated box filter provides for efficient calculation, without adding algorithmic complexity. This allows guided filters to be implemented with large kernels, making them suited for embedded applications with limited resources and real-time requirements. The addition of a pre-filtering stage before decimation improves the quality of the rapid guided filter approximation greatly. The decimation factor affects the PSNR value, which is determined mainly by the size of the filters employed in the decimation domain. The suggested technique works best in the low delay configuration, achieving a 0.67% average BD-rate reduction.

In the future, research is expected to investigate if machine learning techniques can be applied to overcome this problem. It starts with super-resolution CNN (SRCNN), very deep super-resolution (WDSR), and broad activation super-resolution (WASR), all of which were built on convolutional neural networks (CNN). When deep learning was applied to improve in-loop filtering, the focus was primarily on visual enhancement and compression artifact reduction while keeping coding efficiency [29].

4.2. Comparative analysis

The proposed filter performance in VVC architecture is compared with the other filters performance considering the bitrate and encoder time complexity. In contrast to the existing methods, IFGF shows better values. Table 2 shows a comparison of existing techniques with the proposed method.

Table 2. Comparative analysis of proposed and existing methods

Authors	Method	Average BD-Bitrate (%)	Average ΔT (%)
Wennersten *et al.* [17]	Bilateral filter	0.5	--
Strom *et al.* [18]	Bilateral filter in combination with SAO	0.5	--
Chen *et al.* [19]	Adaptive guided filter	0.41	9
Yin *et al.* [20]	Adaptive self-guided filter	0.44	13
Proposed filter	Improved fast guided filter	0.67	4

5. CONCLUSION

Instead of the bilateral filter, which operates efficiently and accurately, the proposed method focuses on an edge-preserving filter. The guided filter is a versatile filter, where new features or models are adopted to obtain new filters. In this paper, an improved fast-guided filter is used in the VVC encoder which is inserted between the deblocking filter and the SAO filter. The algorithm of fast guided filter is modified to overcome the aliasing error due to upsampling. The modified algorithm leads to an improved fast-guided filter by using a prefiltering step. This step uses a low pass filter which involves narrowing the spatial components before performing the decimation by a factor S. The quality of the videos increases with the proposed filter followed by a considerable increase in encoder time. The results are obtained by integrating the C++ program in VTM-17.2 software and the suggested algorithm achieves a +1.007 dB increase in PSNR and 0.67% average BD-rate reduction in comparison to the existing algorithm with the expense of a 4% increase in encoder time. In the future, the algorithm may be optimized to decrease the encoder/decoder complexity time and to enhance the compression efficiency.

ACKNOWLEDGEMENTS

The authors are grateful to the administration of VIT-AP University for allowing us to work in a peaceful environment, without which this work would not be possible.

REFERENCES

[1] "H.266 : versatile video coding," *International Telecommunicarion Union (ITU)*, 2022. https://www.itu.int/rec/T-REC-H.266 (accessed Nov. 16, 2022).
[2] "H.265 : high efficiency video coding," *International Telecommunication Union (ITU)*, 2019. https://www.itu.int/rec/T-REC-H.265 (accessed Nov. 16, 2022).
[3] "H.264 : advanced video coding for generic audiovisual services," *International Telecommunicarion Union (ITU)*, 2019. https://www.itu.int/rec/T-REC-H.264 (accessed Nov. 16, 2022).
[4] B. Bross *et al.*, "Overview of the Versatile video coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021, doi: 10.1109/TCSVT.2021.3101953.
[5] F. Pakdaman, M. A. Adelimanesh, M. Gabbouj, and M. R. Hashemi, "Complexity analysis of next-generation VVC encoding and decoding," in *Proceedings - International Conference on Image Processing, ICIP*, Oct. 2020, vol. 2020-October, pp. 3134–3138, doi: 10.1109/ICIP40778.2020.9190983.
[6] K. Andersson, K. Misra, M. Ikeda, D. Rusanovskyy, and S. Iwamura, "Deblocking filtering in VVC," in *2021 Picture Coding Symposium, PCS 2021 - Proceedings*, Jun. 2021, pp. 1–5, doi: 10.1109/PCS50896.2021.9477477.
[7] M. Karczewicz *et al.*, "VVC in-loop filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3907–3925, Oct. 2021, doi: 10.1109/TCSVT.2021.3072297.
[8] P. L. Amruthavalli and P. Nalluri, "A Review on in-loop filters for HEVC and VVC video coding standards," in *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022*, Mar. 2022, pp. 997–1001, doi: 10.1109/ICACCS54159.2022.9784992.
[9] V. Moji and M. Mathivanan, "A survey on in-loop filters of high efficiency video coding standard," in *3rd International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques, ICEECCOT 2018*, Dec. 2018, pp. 1189–1193, doi: 10.1109/ICEECCOT43722.2018.9001558.
[10] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614–619, Jul. 2003, doi: 10.1109/TCSVT.2003.815175.
[11] A. Norkin *et al.*, "HEVC deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012, doi: 10.1109/TCSVT.2012.2223053.
[12] P. L. Amruthavalli and P. Nalluri, "Performance analysis of deblocking filter in VVC," in *2023 3rd International Conference on Artificial Intelligence and Signal Processing, AISP 2023*, Mar. 2023, pp. 1–5, doi: 10.1109/AISP57993.2023.10134828.
[13] H. Faraji and W. J. MacLean, "CCD noise removal in digital images," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2676–2685, Sep. 2006, doi: 10.1109/TIP.2006.877363.
[14] S. Y. Lee and C. E. Rhee, "Motion estimation-assisted denoising for an efficient combination with an HEVC encoder," *Sensors (Switzerland)*, vol. 19, no. 4, p. 895, Feb. 2019, doi: 10.3390/s19040895.
[15] S. Thomas and A. Krishna, "Impulse noise recuperation from grayscale and medical images using supervised curve fitting linear regression and mean filter," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 28, no. 2, pp. 777–786, Nov. 2022, doi: 10.11591/ijeecs.v28.i2.pp777-786.
[16] S. Dixit and D. Nagaria, "LMS adaptive filters for noise cancellation: a review," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2520–2529, Oct. 2017, doi: 10.11591/ijece.v7i5.pp2520-2529.
[17] P. Wennersten, J. Strom, Y. Wang, K. Andersson, R. Sjoberg, and J. Enhorn, "Bilateral filtering for video coding," in *2017 IEEE Visual Communications and Image Processing, VCIP 2017*, Dec. 2017, vol. 2018-January, pp. 1–4, doi: 10.1109/VCIP.2017.8305038.
[18] J. Strom, P. Wennersten, J. Enhorn, D. Liu, K. Andersson, and R. Sjoberg, "Bilateral loop filter in combination with SAO," in *2019 Picture Coding Symposium, PCS 2019*, Nov. 2019, pp. 1–5, doi: 10.1109/PCS48520.2019.8954554.
[19] C. Chen, Z. Miao, and B. Zeng, "Adaptive guided image filter for improved in-loop filtering in video coding," in *2015 IEEE 17th International Workshop on Multimedia Signal Processing, MMSP 2015*, Oct. 2015, pp. 1–6, doi: 10.1109/MMSP.2015.7340842.
[20] W. Yin, K. Zhang, L. Zhang, Y. Wang, and H. Liu, "Adaptive self-guided loop filter for video coding," in *2021 International Conference on Visual Communications and Image Processing, VCIP 2021 - Proceedings*, Dec. 2021, pp. 1–5, doi: 10.1109/VCIP53242.2021.9675440.
[21] S. Ikonin, V. Stepin, R. Chernyak, and J. Chen, "Hadamard transform domain filter for video coding," in *Applications of Digital Image Processing XLII*, Sep. 2019, p. 37, doi: 10.1117/12.2530513.
[22] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013, doi: 10.1109/TPAMI.2012.213.
[23] K. He and J. Sun, "Fast guided filter," *arXiv preprint*, 2015.

[24] D. El-Mezeni and L. Saranovac, "Fast guided filter for power-efficient real-time 1080p streaming video processing," *Journal of Real-Time Image Processing*, vol. 17, no. 3, pp. 511–525, Jul. 2020, doi: 10.1007/s11554-018-0802-z.

[25] "The VVC test model." https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware VTM/-/releases (accessed Mar. 08, 2022).

[26] H. Amirpour, V. V. Menon, S. Afzal, M. Ghanbari, and C. Timmerer, "VCD: video complexity dataset," in *MMSys 2022 - Proceedings of the 13th ACM Multimedia Systems Conference*, Jun. 2022, pp. 234–239, doi: 10.1145/3524273.3532892.

[27] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012, doi: 10.1109/TCSVT.2012.2221192.

[28] L. Song, X. Tang, W. Zhang, X. Yang, and P. Xia, "The SJTU 4K video sequence dataset," in *2013 5th International Workshop on Quality of Multimedia Experience, QoMEX 2013 - Proceedings*, Jul. 2013, pp. 34–35, doi: 10.1109/QoMEX.2013.6603201.

[29] H. K. Joy and M. R. Kounte, "Deep learning-based switchable network for in-loop filtering in high efficiency video coding," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 4, pp. 4640–4648, Aug. 2023, doi: 10.11591/ijece.v13i4.pp4640-4648.

BIOGRAPHIES OF AUTHORS

Lakshmi Amrutha Valli Pamidi received her B.Tech. degree in Electronics and Communication Engineering (ECE) from Andhra University, Vizag, and M.Tech. degree in VLSI design from Jawaharlal Nehru Technological University Kakinada. Currently, she is a research scholar at the School of Electronics Engineering, VIT-AP University, Andhra Pradesh-522237, India. Her main research areas include image processing and video coding. She can be contacted at email: amruthavalli045@gmail.com.

Purnachand Nalluri obtained his Master's degree from VIT University in India and completed his Doctoral degree at the University of Aveiro in Portugal. Currently, he serves as an Associate Professor at VIT-AP University, India. His research interests include video processing and pattern recognition. Furthermore, he has substantial interests in FPGA and ASIC-based architecture design for video processing and pattern recognition algorithms. He can be contacted at email: chanduinece@gmail.com.