HEVC-BASED MOTION COMPENSATED JOINT TEMPORAL-SPATIAL VIDEO DENOISING

Minhao Tang, Yuxing Han, Jiangtao Wen, Shiqiang Yang

Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China, jtwen@tsinghua.edu.cn

ABSTRACT

A novel HEVC-based efficient video denoising algorithm is proposed in this paper. It uses a spatial Gaussian filter for the chrominance components and then utilizes the HEVC motion estimation process to find the best temporal correspondence for low-pass filtering. Other HEVC tools such as quantization, the interpolation and the in-loop filters are also used. Experiments implementing the proposed algorithm in the open-source HEVC encoder x265 showed a good denoising performance with a much lower computing complexity than the competitors. The performance was comparable to those highly sophisticated algorithms such as the VBM4D, which is 200 times slower. The proposed algorithm can be easily integrated into the real-world video processing systems due to its compatibility with the HEVC standard.

***Index Terms*—** Video Denoising, HEVC, Spatial-Temporal Filter

1. INTRODUCTION

Noise is inherent to the acquisition, processing and transmission of digital videos. For example, the imperfection of the CCD and CMOS sensors as well as the A/D, D/A processes will often introduce noise to the video signal, which causes abnormal variations in spatially neighboring pixels that can be easily noticed and cause perceptual quality degradations. In addition, due to its random nature, video noises will lessen the efficiency of video encoders required for video transmission and/or storage. Therefore, video denoising is a critical challenge for high quality efficient video applications.

The color videos are generally captured using an array of color sensors for the different color channels. The noise generated by the sensors are often considered to be independent of each other, and modeled as an additive zero-mean Gaussian white noise on top of the pixel values of the color channels.

Existing algorithms for video denoising can be roughly divided into pixel domain and transform domain algorithms. Pixel domain denoising algorithms usually rely on similarities between spatially or temporally neighboring pixels. Weighted averaging based spatial and/or temporal low pass filtering is used to reduce the noise while preserving video details. Because the temporal similarity assumption might be invalid after scene changes or for high motion clips, the strength of the temporal filtering must be made adaptive. Adaptive temporal averaging methods usually work by first assessing the level of temporal similarities to determine the pixels whether to be temporally averaged. Examples of adaptive temporal averaging include the ATA denoising [1] and the hqdn3d denoiser in GPLv2, both of which have been incorporated into the well-known open source multimedia tool ffmpeg [2]. To improve the accuracy of temporal averaging by pin-pointing temporal similarities, motion estimation (ME) is usually used to find the best match region in the temporally neighboring frame.

On the other hand, transform domain denoising has become one of the most popular topics in the last decade. Such algorithms would first transform the original RGB signal into another domain. Classification of the features in the transform domain (e.g. maximum likelihood, Bayesian estimation) determines whether and how to denoise the video. The wavelet transform and its various extensions are the most widely used transforms in the transform domain denoising. Such algorithms include over-complete wavelet denoising (owdenoise) in the ffmpeg software, SEQWT [3] and WRSTF [4]. In addition to the wavelet transform, the latest denoising algorithms such as VBM3D [5] and VBM4D [6] designed some other novel transforms and produced a much better performance than SEQWT and WRSTF.

In this paper, we proposed an efficient video denoising algorithm using HEVC [7] video coding tools, especially the highly effective HEVC ME tools for improved temporal-spatial averaging, as well as quantization, the interpolation and in-loop filters. Experiments implementing the proposed algorithm into the open source HEVC encoder x265-v1.9 [8] showed that the proposed algorithm, despite its simplicity, achieved comparable denoising performance when compared with many widely used denoising algorithms, some of which are orders of magnitudes slower.

The remainder of this paper is organized as follows. Section 2 describes the proposed algorithm in detail. Section 3 presents the performance of the proposed algorithm and its comparison with some widely used algorithms. Section 4 concludes the paper.

This work was kindly supported by Nanjing Yunyan Information Technology Ltd and Microsoft.

2. PROPOSED ALGORITHM

Fig. 1 gives the flow diagram of the proposed denoising algorithm, which will be described in detail in the following subsections.

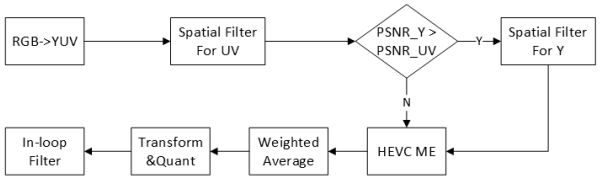

Fig. 1: Flow Diagram of the Proposed Algorithm

2.1. Color Space Transform

The RGB color model is an additive color model in which red, green and blue lights are added together in various ways to reproduce different colors. The three channels are respectively captured, quantized and stored, though there exist a strong correlation and therefore high redundancy among these three channels. To reduce the redundancy, the YUV model was designed for the compression, storage and transmission of the digital videos. In the model, the Y component represents the luminance while U/V are the chrominance components,

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

In addition to removing redundancy, the transform from the RGB to the YUV model can also help reduce the noise. The independent zero-mean white additive Gaussian noises can be filtered by the weighted averaging in the conversion, so that the noisy version of the YUV components ($Y_nU_nV_n$) can be expressed as

$$Y_n = Y + N(0, 0.45\sigma^2), \quad (2)$$
$$U_n = U + N(0, 0.39\sigma^2), \quad (3)$$
$$V_n = V + N(0, 0.43\sigma^2), \quad (4)$$

where YUV represent the noise-free input and σ is the standard deviation of the zero-mean white additive Gaussian noise in the RGB domain. $N(\mu, \sigma^2)$ is a Gaussian signal with the expected value as μ and the variance as σ^2. As can be seen in Equation 4, the noise level in the YUV space is much lower than that in the RGB space.

The reduction in noise level in the RGB to YUV conversion improves the accuracy of the ME process in subsequent processing, which allows for improved denoising results. Even though the YUV signal will need to be converted back to the RGB space, a process that increases the noise deviation, due to the improved denoising performance, the overall end-to-end quality is still improved.

2.2. HEVC Based Denoising

Due to their low energy levels, in real-world video applications, the UV channels are usually down-scaled to 1/4 of the spatial resolution of the Y channel before processing, resulting in the YUV420p format of digital image or video. When the information contained in the UV components is very limited with the spatially neighboring pixels highly correlated, a simple spatial filter (e.g. the Gaussian filter) can already achieve good denoising results for the UV components with relatively low complexity. To verify this insight, experiments employing the down-sampling and Gaussian filter on the YUV components of YUV444p video clips were conducted. The clips were downloaded from [9] and contained a fairly wide range of different characteristics (i.e. different levels of texture).

According to the results in Table 1, the filtered and down-sampled videos could still maintain a high Peak signal-to-noise ratio (PSNR) value (around 38dB, much higher than the PSNRs for the noisy videos, which are generally around 25dB) for the UV components, while the distortion for the Y component is much greater. From the results, it seems that a 3×3 Gaussian filter is sufficient to maintain a high UV fidelity while simplifying the denoising process. Due to the high distortion caused by the missing details, the Y component should generally be processed in full resolution.

Because the Y channel contains most of the visual information, details in the Y component frames must be preserved while effectively removing noise. But the spatial filter on the Y component can still sometimes help reduce the noise with limited loss in detail when the original details have already been destroyed by the strong noise. To this end, we proposed a noise estimation based adaptive spatial filtering process for the Y frames. A weak spatial filter heavily centered around the current pixel (corresponding to a weight of 60%) will be attempted periodically (1s in the experiment) or when a scene change is detected. The PSNR is calculated between the frames before and after the filtering for the YUV components. Table 1 proves that the PSNR value of the Y channel after being filtered by the same filter should be much lower than the U/V due to the rich details in the Y channel. Therefore, the level of texture can be considered low when the PSNR value of the Y component filtered by a weak filter is greater than the PSNR values of UV channels. In that case, this weak spatial filter will be used for the Y channel frames in the next period.

Table 1: Information of YUV Components

Method	Down-Sampling			3x3 Gaussian		
PSNR	Y	U	V	Y	U	V
IntoTree	32.82	38.34	40.74	33.87	39.14	41.30
OldTown	32.18	38.44	40.53	33.07	39.24	41.11
ParkJoy	25.27	34.02	37.73	26.31	34.92	38.59
Ducks	27.38	34.00	38.79	28.46	35.04	39.70

After the adaptive spatial filtering, HEVC ME is used to find the best match for each block for the subsequent temporal filtering. As the latest and state-of-the-art video coding standard, HEVC has the most advanced set of motion estimation tools for identifying block based temporal correspondences. The HEVC ME introduces a well-studied 7-tap interpolation filter for conducting sub-integer precision motion estimation and compensation. The interpolation filter can also improve the signal to noise ratio as the values of neighboring pixels are correlated while the neighboring noise samples are much less so, if not completely independent. Through experiments we found that the block-distortion based ME can achieve better results than feature based ME algorithms like optical flow based object tracking [10]. This is partially due to the fact that any structural feature can be easily distorted by the noise, rendering feature based algorithms undesirable. In our experiments, the ME distortion is defined as

$$\begin{aligned} D &= \sum_{pixel\ i} (s_i + n_i - r_i - rn_i)^2, \\ &= \sum_{pixel\ i} (s_i - r_i)^2 + \sum_{pixel\ i} (n_i - rn_i)^2 \\ &\quad + 2 \sum_{pixel\ i} (s_i - r_i)(n_i - rn_i), \end{aligned} \tag{5}$$

where s, r, n, rn, D are the noise-free block to be encoded, the noise-free reference block, the noise of the current block, the noise of the reference block and the distortion using the current motion vector respectively. Since n and rn are zero-mean and independent from each other, the expectation of the distortion

$$E(D) = D_{org} + \sigma_n^2 + \sigma_{rn}^2, \tag{6}$$

where $D_{org}, \sigma_n^2\ and\ \sigma_{rn}^2$ are the distortion if noise-free, the variance of n and the variance of rn respectively. As the distortion is the sum of a constant value, a Gaussian signal and a chi-square signal, there is a high probability for the ME process to find the best or an approximately best correspondence (reference) block. In contrast, those structurally different reference blocks will not be chosen due to the high D_{org} value.

The HEVC ME process would recommend a best candidate for each reference frame. Due to scene changes, inappropriate ME settings such as the limited search range, as well as noise, those candidates may not be of sufficient quality from time to time. To mitigate this problem, the rate-distortion cost based scene change detection algorithm in x265 was used to reject references from a different scene. In addition, the noise and the limited search range may lessen the accuracy of the ME. The statistics of the distortions of the candidates is made, so that a candidate with a distortion 20% (a well-tuned threshold) higher than the statistical average will be excluded from the subsequent weighted averaging. Since the different scenes sometimes might own completely different characteristics, the statistics will be cleared at the beginning of each scene and then the new initial value will be set as the average distortion caused by the spatial filter above. After the elimination, the remaining candidates will be assigned with the weight (W) calculated using the following formula for the weighted averaging,

$$W = \frac{1}{D \cdot \Delta T}, \tag{7}$$

where D and ΔT are the distortion of the candidate and the difference in display time between the current frame and the reference frame. The candidates of higher distortion or longer temporal distance ought to have a lower correlation with the current block and therefore a smaller weight. The weight of the current block is the sum of the reference weights to avoid over-aggressive denoising.

After the weighted averaging, the HEVC encoding will proceed to transform, quantization and the in-loop filters including the deblocking filter and the sample adaptive offset (SAO) filter [11]. Because the energy of natural video signals is usually clustered in low frequency coefficients, the quantization coefficient in HEVC is coarser for high frequency components. In contrast, for Gaussian noise, energy will be fairly uniformly distributed among the different frequencies. The use of the HEVC quantization will lead to significant decrease of the strengths of the high frequency components of the noise signal, while doing little damage to the visual signal.

The HEVC common test condition [12] requires four QP values (22, 27, 32, 37) to be tested, with a QP value less than 22 representing good quality. In the proposed denoising algorithm, the QP value first needs to be small enough to preserve the low frequency visual information, but less high frequency noise will be filtered in quantization with the QP value growing smaller. Through the massive experiments, QP=20 provides the best tradeoff between the visual distortion and the noise reduction among the QP values from 15 to 22. After quantization, the deblocking and the SAO filters are applied to repair and smoothen discontinuities between pixels and blocks to improve the overall visual quality.

3. EXPERIMENTS RESULTS

To evaluate the denoising performance and efficiency of the proposed algorithm, we implemented the algorithm in the latest x265 HEVC encoder v1.9. Five CIF (352×288) sequences frequently used in the papers on video denoising were tested. Each input clip was contaminated with three levels of zero-mean white additive Gaussian noise, with the standard deviation set at 10, 15 and 20 as most of the denoising publications did. Besides the proposed algorithm, several other state-of-art denoising algorithms were also tested for comparisons, including hqdn3d and owdenoise in ffmpeg, the new and high quality VBM4D, as well as the algorithms in commercial software or software packages such as NeatVideo

Table 2: PSNR Values of Videos Denoised Using the Proposed Algorithm and the Competitors

Videos	Foreman (352x288)			Garden (352x240)			Miss America (360x288)			Salesman (352x288)			Tennis (352x240)			AVG	FPS
σ	10	15	20	10	15	20	10	15	20	10	15	20	10	15	20		
hqdn3d	33.27	29.30	26.66	33.09	29.23	26.64	33.37	29.34	26.68	33.27	29.30	26.67	33.23	29.29	26.66	29.73	**300**
owdenoise	33.63	29.77	27.10	33.47	29.68	27.05	33.68	29.80	27.11	33.60	29.76	27.09	33.56	29.74	27.08	30.14	4.2
VBM4D	**41.34**	**39.53**	38.27	**36.01**	**34.99**	**34.04**	43.50	42.34	41.30	**40.31**	**38.30**	36.76	37.54	35.20	33.31	**38.18**	0.5
NeatVideo	40.29	39.49	**38.42**	30.11	29.89	29.56	**44.22**	**42.76**	**41.40**	38.66	38.01	**37.18**	35.60	35.07	**34.97**	37.04	25
MSU	38.81	36.72	35.48	31.93	30.07	28.75	41.17	39.12	37.85	38.91	36.82	35.27	36.88	35.07	33.72	35.77	80
Proposed	**39.61**	**37.84**	**36.48**	**35.90**	**33.78**	**32.10**	**43.31**	**41.49**	**39.04**	**39.66**	**37.44**	**36.03**	**38.42**	**35.75**	**33.51**	**37.36**	**100**

(a) original image frame (b) noisy frame ($\sigma = 20$) (c) Proposed

(d) VBM4D (e) MSU (f) NeatVideo

Fig. 2: Denoising results of "Garden" sequence corrupted with a noise, standard deviation σ = 20.

(employed by Premiere, After Effect and FinalCut) v4.2 released in July 2016 [13] and the MSU Denoiser v2.5.1 [14].

The denoising performance is measured by the PSNR value between the denoised video and the original noise-free video in the YUV domain. All algorithms ran on a single thread of an Intel Core i3-4170 CPU running the Windows 10 operating system with a RAM of 8 GB without the GPU acceleration. Only the key HEVC tools relevant to the denoising task were used, including the fractional pixel level ME, quantization and in-loop filters, while the other time-consuming tools like the quad-tree structure, multiple partitions and RDOQ were disabled.

Table 2 gives the denoising performances of these algorithms. The results of the proposed algorithm and the best performance for each test case (i.e. each column) are displayed in bold font. The VBM4D method (written in C language) could always produce the best or second-best results, but ran at an extremely low speed of 0.5 frames per second (fps), for the small resolution CIF video. It would require nearly 20 minutes to denoise a 1-second 1080p clip, or about 1/1200 real time speed, making it infeasible for real-world applications even if SIMD optimizations were introduced. On the other hand, NeatVideo and MSU denoiser were much more efficient than VBM4D, but their performances were highly dependent on the characteristics of the input. For example, NeatVideo ranked first for "Miss America" but bottom for "Garden".

In contrast, the proposed algorithm showed a robust and efficient performance, which is better and faster than NeatVideo and MSU, with a quality close to VBM4D but ran 200 times faster. Fig. 2 illustrates the effectiveness of the proposed denoising algorithm for sequence "Garden" contaminated by a noise with the standard deviation set as 20. There was no longer visible noise in the denoised image after using the proposed denoising algorithm and most of the details were well maintained, while the trunks in Fig. 2-(e)(f) completely lost their textures and became blurry, which was probably caused by the over-aggressive temporal averaging and the inaccurate ME process.

4. CONCLUSION

In this paper, we proposed an efficient video denoising algorithm using the HEVC motion estimation based temporal-spatial averaging, quantization and filters to improve the quality of denoising while achieving a high processing speed. Experiments implementing the proposed algorithm in the open-source HEVC encoder x265 show that the proposed algorithm produced a denoising quality that is close to high quality but very slow denoising algorithms such as VBM4D, and superior to other fast denoising algorithms such as NeatVideo and the MSU denoiser, while at the same time achieving a speed that is two magnitudes faster than VBM4D. Given that the tools used in the denoiser are widely optimized in the HEVC encoding context, the proposed algorithm could be easily integrated into real-world video processing systems.

5. REFERENCES

[1] David Bartovčak and Miroslav Vrankić, "Video denoising based on adaptive temporal averaging," *Engineering Review*, vol. 32, no. 2, pp. 64–69, 2012.

[2] "ffmpeg," http://ffmpeg.org/.

[3] Aleksandra Pizurica, Vladimir Zlokolica, and Wilfried Philips, "Combined wavelet domain and temporal video denoising," in *Advanced Video and Signal Based Surveillance, 2003. Proceedings. IEEE Conference on*. IEEE, 2003, pp. 334–341.

[4] Vladimir Zlokolica, Aleksandra Pizurica, and Wilfried Philips, "Wavelet-domain video denoising based on reliability measures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 8, pp. 993–1007, 2006.

[5] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[6] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," *IEEE Transactions on image processing*, vol. 21, no. 9, pp. 3952–3966, 2012.

[7] Gary J Sullivan, Jens Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the high efficiency video coding (hevc) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.

[8] "x265," www.videolan.org/developers/x265.html.

[9] "xiph.org," https://media.xiph.org/video/derf/.

[10] Berthold KP Horn and Brian G Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[11] Chih-Ming Fu, Elena Alshina, Alexander Alshin, Yu-Wen Huang, Ching-Yeh Chen, Chia-Yang Tsai, Chih-Wei Hsu, Shaw-Min Lei, Jeong-Hoon Park, and Woo-Jin 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.

[12] F Bossen, "Common hm test conditions and software reference configurations, document jctvc-l1100," *JCT-VC, Geneva, Switzerland*, 2013.

[13] "Neat Video," https://www.neatvideo.com/.

[14] "MSU Denoiser," www.compression.ru/video/denoising.