Real-Time Noise Level Detection for General Video

Chinatsu Mori, and Seiichi Gohshi

Kogakuin University
1-24-2 Nishi-Shinjuku, Shinjuku-ku, Tokyo, Japan
Email: gohshi@cc.kogakuin.ac.jp

***Abstract*—Currently, 4K TV is a standard TV and 8K broadcasting has began in December 2018. High-resolution in conjunction with low noise is an essential figure of merit in video systems. Unfortunately, any increase in resolution unavoidably increases noise levels. A signal processing method called noise reducer (NR) is often used to reduce noise. However, accurate noise level is needed when NR is employed. Noise level depends mostly on lighting conditions and is estimated by comparing adjacent frame difference. However, the frame difference is generated by moving objects as well as noise. Therefore, it is essential to determine whether the frame difference is caused by the moving objects or by the noise, which is a difficult task. Another difficulty arises from the fact that noise level detection must be achievable in real-time conditions since all video systems are required to work in real-time. This means that a complex method could not be used for noise level detection. In this paper, two noise level detection algorithms are presented. The combination of two of them is a concise algorithm able to accurately detect the noise level and work in real-time conditions.**

Keywords–Video noise reducer; 4KTV; 8KTV; Real-time; Nonlinear signal processing; Image quality.

I. Introduction

A dramatic change in imaging technologies has taken place in the 21st century. High Definition Television (HDTV) broadcasting started only 20 years ago and at that time HDTV sets were expensive. Today, HDTV is already a part of history. 4K TV broadcasting started a couple of years ago and 8K satellite broadcasting is started in December 2018. Although significant advances have been made in video resolution, imaging technologies are based on the same principle, i.e., the photoelectric effect. Imaging devices primarily comprise of photoelectric cells and the number of electrons generated by each cell is proportional to the number of photons received by the cell. As the resolution increases from HDTV to 4K and then to 8K, the size of the image cell decreases, i.e., the number of photons per image cell is inversely proportional to the resolution. Therefore, it is necessary to amplify the electric energy of a video signal at the output of a video camera.

The electrical energy generated by the image cell is amplified by a pre-amplifier for each pixel. An amplifying process always results in thermal noise called “Gaussian noise.” The level of noise is inversely proportional to the electric energy generated per cell. This is because fewer photons generate a lower voltage signal that requires amplification to achieve the appropriate voltage level. As HDTV, 4K, and 8K are high-resolution systems, the noise level increases because the size of the image cells becomes smaller due to the high-resolution. The best way to reduce noise in a high-resolution video is to increase the sensitivity of image cells’ photoelectric effect. However, in order to achieve this, there are technical limitations, which need to overcome. Even high-end mature HDTV cameras may have pulse noise called “Shot noise” under poor lighting conditions, such as night time shooting or shooting in a dark room.

Noise reducer (NR) is a technology able to reduce noise in video systems by using signal processing techniques. Although, a large number of NR algorithms have been reported most of them are complex and only compatible with still images. The use of such an algorithm in real-time video systems would cause a video to freeze. In other words, complex NR algorithms are not suitable for use in real-time video systems. Another issue is the ability to detect accurate noise levels in video/image systems before applying noise reducing techniques. In case of real-time video systems, noise levels should be detected in real-time as well. Adjacent frame difference is a basic method to detect noise levels. However, noise, as well as moving objects, is contained in the frame, which makes the detection of accurate noise levels in a real-time video a difficult task. In this paper, a real-time noise level detection method is proposed.

This paper is organized as follows. In Section II, related works of NR and noise level detection are explained. In Section III, two noise level detection algorithms are proposed. In Section IV, simulation results are presented. In Section V, the advantages and disadvantages of the algorithms are discussed and the combination of two of them is investigated. Finally, in section VI, conclusions of this work are presented.

II. Related Works

Conventional NR uses spatial or temporal digital filter to reduce noise [1]–[5]. Many NR methods are used for still images. They are spatial digital filters. Generally, the spatial digital filters cause image blurring. Although the common method is NR with wavelet transformation [6]–[9], the application of this method in videos is difficult: because real-time performance is required. Hence, an NR with a recursive temporal filter [10] is the only practical real-time method used for videos. However, it is necessary to know the accurate noise level for the NR to work. Generally, videos comprise a wide variety of content with different noise levels. The differences are also caused by lighting conditions. In the development of automatic, real-time NR hardware, the NR parameter must be set properly in accordance with the actual noise level of a video. Although the adjacent frame difference is the basis of noise level detection, the frame difference is the result of noise and the moving areas.

Only a few proposals for noise level detection methods in videos are available. The wavelet transformation is used

 ISBN: 978-1-61208-707-8

for the noise level detection [11], [12], but its real-time work application is difficult owing to its high processing cost.

The spatial and temporal digital filter is simple and is used for noise level estimation with low cost [13], [14]. Gaussian noise can be detected by applying high-pass filter, such as Sobel filter and Laplacian filter. However, these filters detect both noise and temporal moves of videos: the noise level is overestimated if the video includes fast and complex moves, such as camera works and object moves.

In the authors' previous works, a noise level detection method which uses a bilateral filter has been proposed [15]. However, the bilateral filter also comes with a high hardware cost. A noise level detection algorithm is essential not only for the real-time function but also the accurate determination of the actual noise level. The method that uses the bilateral filter fails to perform when the noise level is high. Therefore, some improvements are necessary to address these issues.

III. Proposed Methods

In this paper, two noise level detection algorithms are proposed and the combination of these methods is considered.

A. Noise Level Estimation

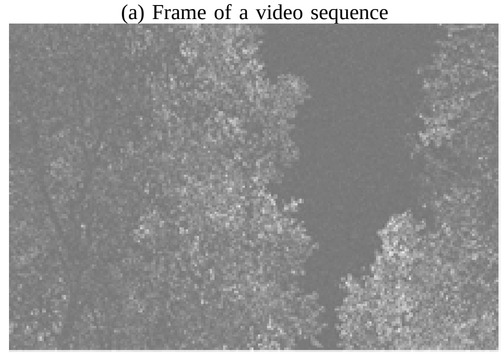
A video has three axes, namely, vertical, horizontal, and the frame. The plane that consists of the vertical and horizontal axes is called spatial, whereas the frame axis is called temporal. By comparing the correlations of spatial and temporal, the spatial correlation is stronger than the temporal. The conventional NR [10] uses the temporal characteristic, as does the noise level detection algorithm. However, the adjacent frame difference is the most effective method to detect the noise level, but it involves two types of signals: frame differences caused by noise and that by moving objects in a video.

Figure 1 illustrates some examples. Figure 1 (a) presents the frame of a video [16]. In the sequence, trees and leaves rustle in the wind. Figure 1 (b) shows the frame difference caused by the trees and leaves. The noise level can be obtained by the standard deviation of the frame difference values in the flat areas because the frame difference in the flat areas is created by noise. Thus, separating the flat areas with frame difference caused by noise from the areas with moving objects is necessary. There are two characteristics of the frame differences for separating the flat areas and moving areas. The frame difference caused by moving objects has shapes and areas, whereas that caused by noise is isolated. Moreover, moving objects have large frame difference values, whereas noise often generates small difference values. Based on these characteristics, we introduce two NR methodologies.

B. Frame Difference and Threshold Process

As discussed in Section III-A, the frame difference values caused by the moving objects are larger, thus, distinguishing these two using a threshold process is possible. Figure 2 shows the block diagram of the noise level detection with frame difference and threshold processing. The frame difference is detected using a frame memory and the input frame. In the threshold processing, only a small frame difference is selected, and its values and pixel numbers are sent to the noise level calculation block. In the noise level calculation block, the frame difference values and pixel numbers are accumulated. The average noise level can be measured using these two values. Figure 3 shows the candidate of the flat areas using the frame difference and threshold process. However, this method also incorrectly identifies the frame difference caused by moving objects in the tree areas. The moving objects do not always produce large frame difference values. With the luminance-level difference between the moving objects and the background, the frame difference values are small and can sometimes generate similar values to those caused by noise. Although the frame difference between the blue sky and the trees is substantial in the video shown in Figure 1 (b), the frame difference among the tree leaves is minimal and similar to the values caused by noise. The incorrect identification due to similar magnitudes in change between moving objects and noise is the problem with this method.

(a) Frame of a video sequence



(b) Frame difference of (a)

Figure 1. Video frame and frame difference

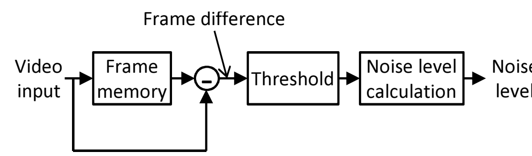


Figure 2. Frame difference and threshold process

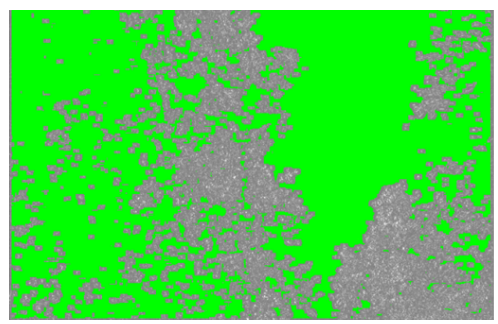


Figure 3. Areas detected using the frame difference and threshold process

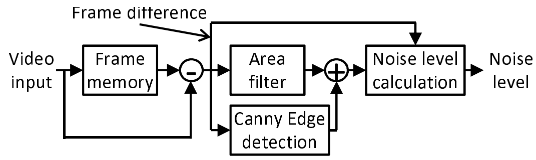
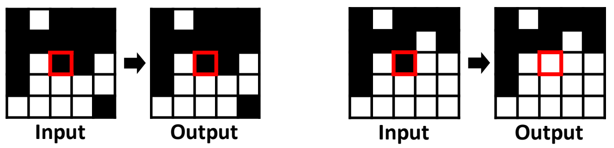


Figure 4. Proposed method 1



(a) Interest pixel as flat area (b) Interest pixel as moving area

Figure 5. Examples of area filter process.

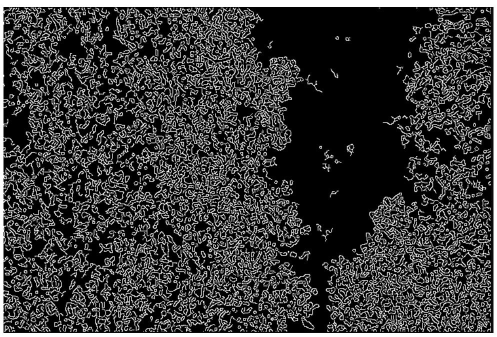


Figure 6. Output of the Canny edge detection block

C. Proposed Method 1: Area Filter and Edge Detection

As shown in Figure 3, the frame difference caused by the tree leaves is detected as the flat areas for determination of the noise level. Although the moving objects, that is, trees and leaves, result in large frame difference values, some can be quite similar to the nearby areas, such as white shining leaves and the blue sky. The shining leaves and the sky produce small frame difference, such as noise because they have similar luminance levels. To prevent this issue, we need to connect these areas and exclude the spaces from analysis. Thus, we introduce the area filter and Canny edge detection [17] illustrated in Figure 4, to improve the noise level accuracy.

Based on the input to the frame difference detection, Figures 4 and 2 are similarly presented. The frame difference is distributed into three blocks: the area filter, the Canny edge detection, and the noise level detection in Figure 4. The function of the area filter is illustrated in Figure 5, and is a symmetric nonlinear type of filter. The center pixel value is processed with the surrounding pixel values and has two parameters, the kernel size and the threshold level. The kernel size is 5×5, as shown in Figure 5. The input of the area filter is the frame difference and has positive and negative values.

In the area filter block, the frame difference is processed with an absolute function to render all values positive. The absolute values are identified using the algorithm presented in Figure 5. The white pixels indicate values exceeding the threshold level, whereas the black pixels are equal to or less than the threshold level. If the number of the surrounding pixels exceeding the threshold level is the majority, the area filter decides the interest pixels as the moving area, otherwise, it decides the interest pixels as the flat area. As shown in Figure 5 (a), the number of pixels exceeding the threshold is 11 (the white bocks), and the number equal to or less than the threshold is 14 (the black blocks). In this case, the output of the center pixel is as the flat area. As shown in Figure 5 (b), the number of pixels exceeding the threshold is 14 (the black blocks) and less than or equal to the threshold is 11 (the white blocks). Therefore, the output of the center pixel is as the moving area. By using the following method, we can detect most of the moving areas in Figure 3, but not quite all of them. Therefore, we also introduce the Canny edge detection. The Canny edge detection identifies the continuous edges in the frame difference. These edges are caused by the leaves. A couple of pixels around the Canny detected edges are obtained from the Canny edge detection block. The result of the Canny edge detection is shown in Figure 6. By using the logical OR on the area filter and edge detection blocks, the appropriate areas for the noise level detection are accurately detected.

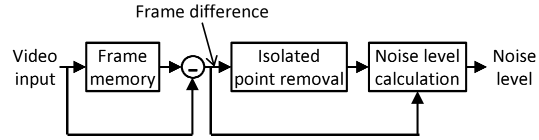


Figure 7. Proposed method 2-A

(a) Interest pixel in moving areas (b) Interest pixel in flat areas

Figure 8. Examples of isolated point removal process

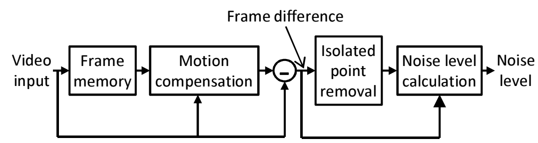


Figure 9. Proposed method 2-B

D. Proposed Method 2: Isolated Point Removal and Motion Compensation

The proposed method 1 shown in Figure 4 can accurately detect the noise level when standard deviation is less than 9. We will discuss the problem in the following section in detail. To address the problem that arises when standard deviation is higher than 9, we have proposed another method.

The signal flow of the proposed method 2-A for a high-level noise is shown in Figure 7. The frame difference detection process of the input and frame memory blocks in Figure 7 is the same as that in Figure 4. The frame difference is distributed into two blocks. The first one is the isolated point removal block and the other one is the noise level calculation block. As discussed in Section III-A, the frame differences are caused both by moving objects and noise. Given that noise level can be detected in flat areas, discriminating the flat areas with noise from the entire frame is necessary. Generally, the frame differences caused by noise in flat areas are isolated. When isolated point removal is used, the output of the isolated point removal block can be the same as the frame difference caused by the moving object, and the noise level can be estimated

 ISBN: 978-1-61208-707-8

using the areas excluding the detected moving areas.

The isolated point removal process is shown in Figure 8. The center pixel in Figure 8 is the interest pixel. Figure 8 (a) shows an example where the interest pixel is the moving area, and Figure 8 (b) illustrates the noise on the flat area. The input is the frame difference. Moreover, the absolute value of the frame difference is calculated and is binarized using the threshold level. The pixels shown in Figure 8 are the result of the binarization. The black areas are below or equal to the threshold level, indicating the flat area. Meanwhile, the white areas are higher than the threshold level, which are candidates similar to the moving areas or the noise on the flat areas. Using only the flat areas is necessary for the noise level estimation. Thus, in the isolated point removal process, the candidate pixels in the white areas are removed if the pixel is isolated and identified as the noise on the flat area. The parameter of the pixel size of the noise is used and the pixel size is set to 5 pixels, as shown in Figure 8. As presented in Figure 8 (a), the pixel size of the white area contains 7 pixels, which is larger than 5. The process identifies the area to be the moving area. As shown in Figure 8 (b), the pixel of the white area contains 4 pixels, which is less than 5. In this case, the pixel is determined to represent the noise, and it is removed.

Many frame differences are present in the frames. These differences have larger values when a video includes camera works, such as panning and tilting. However, the threshold process cannot detect the frame difference accurately for the noise level detection. Thus, we also introduce a block-based motion compensation to detect and reduce moving areas in the frame difference. The proposed method 2-A with motion compensation (method 2-B) is shown in Figure 9. The process of the motion compensation block; the frame is partitioned into blocks of pixels, and each pixel of a block is shifted to the position of the predicted block via the motion vector. This process is common in the discussions of video coding technologies, such as MPEG-2, MPEG-4, and HEVC. Furthermore, we verify and discuss the performance of the motion compensation in the following sections.

IV. EXPERIMENT

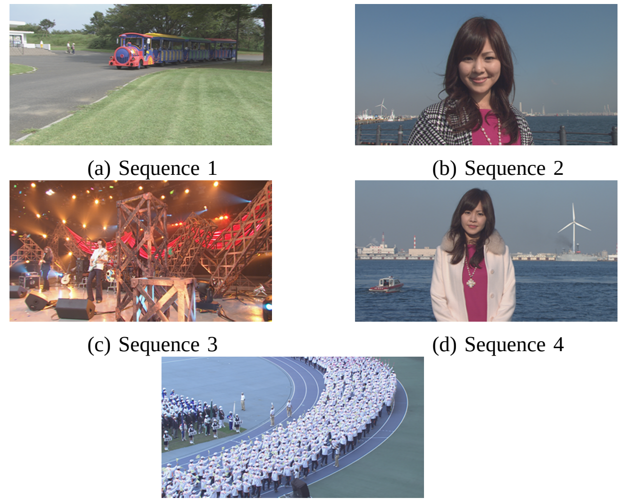
Simulation experiment was conducted to verify the performance of the proposed methods. Different levels of noise were added to video sequences, and the accuracy of the estimated noise level determined by each method was compared.

A. Test Sequences

Noise levels in general videos were estimated using the frame difference (Section 3.1), the proposed method 1 (Section 3.3), and the proposed methods 2-A and 2-B (Section 3.4). The five HDTV $(1,920 \times 1,080)$ video sequences [16] shown in Figure 10 were used in this experiment. All sequences included moving objects and various camera actions, such as panning and tilting. Gaussian noise with different standard deviations (1, 3, 5, 7, 9, 11, 13, and 15) was added to the videos.

B. Experimental Results

The experimental results are shown in Figures 11 and 12. Figures 11 (a)-(e) show the results for sequences 1-5 respectively. The figures show the estimated standard deviation for each level of added noise. The x-axis is the standard deviation of the noise added to the test sequence, and the y-axis



(a) Sequence 1 (b) Sequence 2

(c) Sequence 3 (d) Sequence 4

(e) Sequence 5

Figure 10. Test sequences

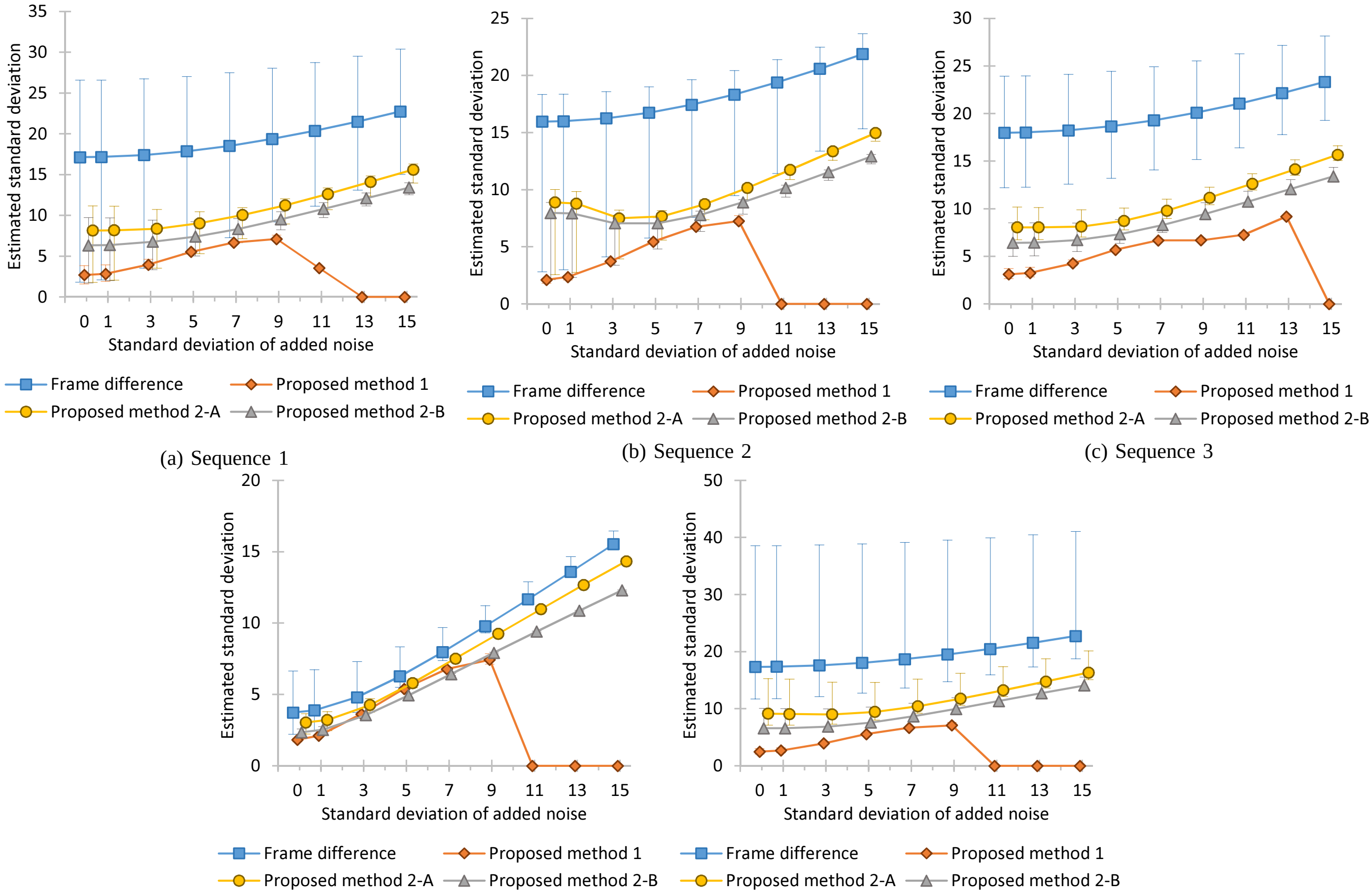
is the estimated standard deviation of the noise in the sequence. The marks show the median values of the estimated standard deviations. If the estimated noise level is correct, the result has the same value as the added noise standard deviation, i.e., $y = x$. The bars indicate the minimum to maximum range of the estimated noise standard deviation, which shows the variation of the results in the sequence.

In Figures 11 (a)-(e), the results for the frame difference method are overestimated and demonstrate large variance. The estimated results for the proposed method 1 are the most accurate and have the smallest dispersion of results. However, the estimation is not possible with the noise standard deviation exceeding 9 because there are few or no appropriate areas for calculating noise standard deviation. The proposed methods 2-A and 2-B returned fewer errors and demonstrate more consistent estimated results than the frame difference. However, large errors tend to occur when the noise standard deviation is less than 3. A comparison of the results for the proposed methods 2-A and 2-B, with and without motion compensation, demonstrates that motion compensation is effective in certain cases. However, it increases the cost significantly because a real-time motion compensation requires large hardware.

Figure 12 shows the estimated noise standard deviation for all frames of sequence 1 (Figure 11 (a)). Figures 12 (a) and (b) show the estimation results for the proposed methods 2-A and 2-B when the noise level is larger than 9. Here the x-axis is the frame number, and the y-axis is the estimated standard deviation of the noise in the frame. The results become constant if the noise level estimation is correct.

In sequence 1, the train is moving with camera panning from 0 to 150 frames, then the panning stops. The train continues to move during frames 150 to 420. There is no motion in frames 420-450. As shown in Figures 12 (a) and (b), the effect of motion on the estimation result is negligible, and the results become constant.

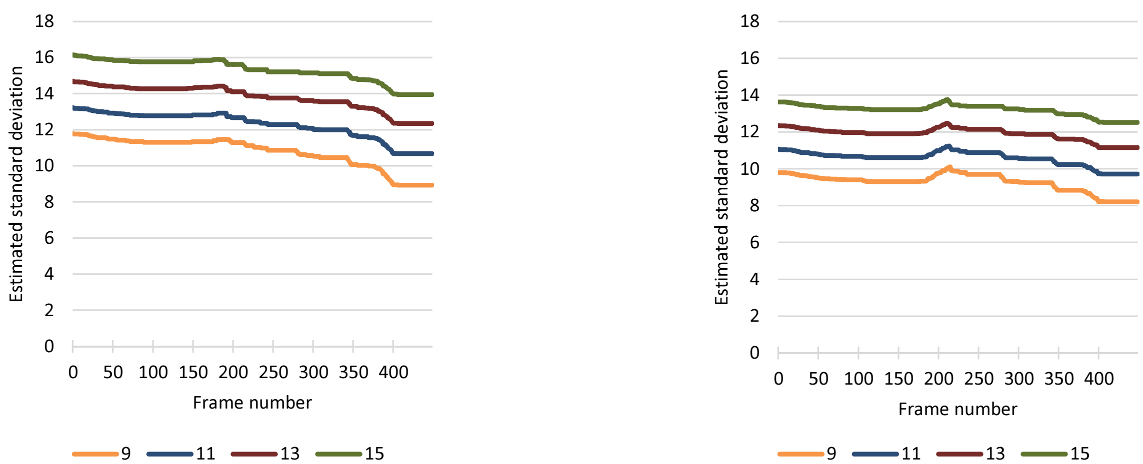
Comparisons of the areas for noise estimation using the proposed method 1, and the proposed method 2-A are shown in Figure 13. The estimated noise areas for sequence 1 with added Gaussian noise are shown in Figures 13 (a)-(b) (standard deviation 3) and Figures 13 (c)-(d) (standard deviation 7). Here, the white areas are estimated moving areas; thus, only



(a) Sequence 1 (b) Sequence 2 (c) Sequence 3

(d) Sequence 4 (e) Sequence 5

Figure 11. Results of estimated noise standard deviation. (a) - (e) show the results for sequences 1-5, respectively. The estimated results of all frames of the video sequence are accumulated. The marks show the median values of the estimated standard deviations. The bars indicate the maximum and minimum values.

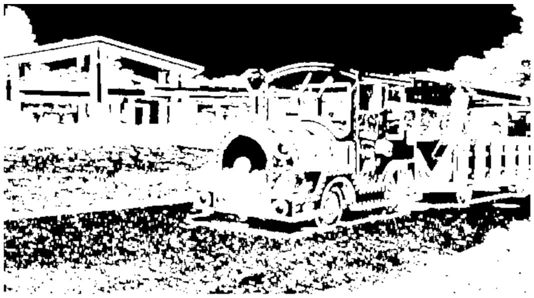


(a) Proposed method 2-A (b) Proposed method 2-B

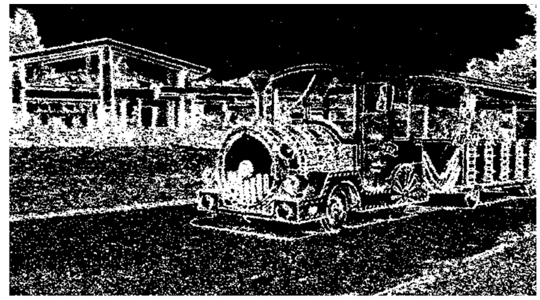
Figure 12. Results of estimated noise standard deviation in time axis for sequence 1 using the proposed methods (a) 2-A and (b) 2-B

the black areas are used for noise estimation. When comparing Figure 13 (a) with Figure 13 (b), and Figure 13 (c) with Figure 13 (d), the moving areas estimated using the proposed method 1 are thick; however, there are few areas for noise estimation when the noise level is high. Since the proposed method 1 fully eliminates moving areas, the noise level estimation becomes accurate. However, the noise estimation does not work with high level noise due to few or no available estimation areas. In contrast, the estimated moving areas using the proposed method 2-A are thin; therefore, the moving areas of the frame with high level noise are detectable.

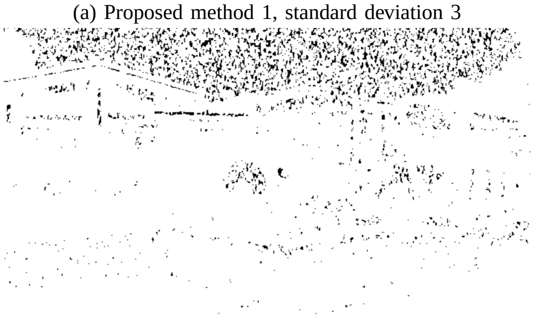
 ISBN: 978-1-61208-707-8



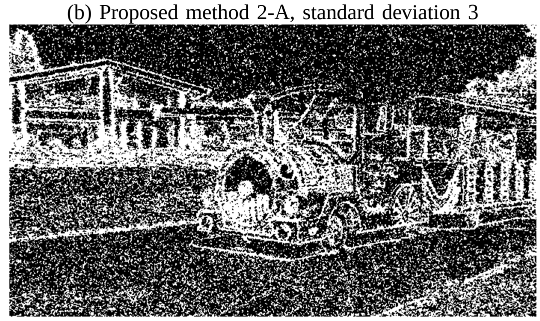
(a) Proposed method 1, standard deviation 3



(b) Proposed method 2-A, standard deviation 3



(c) Proposed method 1, standard deviation 7



(d) Proposed method 2-A, standard deviation 7

Figure 13. Areas of calculated noise standard deviations for one frame of sequence 1

V. Discussion

As described in Section IV-B, the proposed method 1 can detect low level noise accurately when the standard deviation is less than 9. However, this method requires improvement to detect high level noise when the standard deviation is higher than 9. In contrast the proposed methods 2-A and 2-B can detect high level noise when the standard deviation is 5 or more. Therefore, we propose combining the proposed methods 1 and 2-A, i.e., when the detected noise level is less than 9, the proposed method 1 is appropriate and when the detected noise level is equal to or higher than 9, the proposed method 2-A is appropriate. Moving compensation can improve noise detection accurately; however, it requires significantly more expensive hardware.

VI. Conclusion

In this paper, real-time noise level detection algorithms for videos were proposed. The simulation results demonstrate that the best results can be realized by combining two methods. In future, we intend to develop a way to switch between methods automatically and to control NR using the proposed methods. Ultimately, we hope to develop real-time noise reduction hardware that controls noise level parameters automatically.

References

[1] M. Kazubek, “Wavelet domain image denoising by thresholding and wiener filtering,” *IEEE Signal Processing Letters*, vol. 10, no. 11, pp. 324–326, 2003.

[2] A. Pizurica, V. Zlokolica, and W. Philips, “Noise reduction in video sequences using wavelet-domain and temporal filtering,” in *Wavelet Applications in Industrial Processing*, vol. 5266. International Society for Optics and Photonics, 2004, pp. 48–60.

[3] N.-X. Lian, V. Zagorodnov, and Y.-P. Tan, “Video denoising using vector estimation of wavelet coefficients,” in *2006 IEEE International Symposium on Circuits and Systems*. IEEE, 2006, pp. 2673–2676.

[4] I. W. Selesnick and K. Y. Li, “Video denoising using 2d and 3d dual-tree complex wavelet transforms,” in *Wavelets: Applications in Signal and Image Processing X*, vol. 5207. International Society for Optics and Photonics, 2003, pp. 607–619.

[5] A. Pizurica, V. Zlokolica, and W. Philips, “Combined wavelet domain and temporal video denoising,” in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003*. IEEE, 2003, pp. 334–341.

[6] N. Gupta, M. Swamy, and E. I. Plotkin, “Low-complexity video noise reduction in wavelet domain,” in *IEEE 6th Workshop on Multimedia Signal Processing, 2004*. IEEE, 2004, pp. 239–242.

[7] R. O. Mahmoud, M. T. Faheem, and A. Sarhan, “Comparison between discrete wavelet transform and dual-tree complex wavelet transform in video sequences using wavelet-domain,” *INFOS2008*, pp. 20–27, 2008.

[8] L. Jovanov, A. Pizurica, S. Schulte, P. Schelkens, A. Munteanu, E. Kerre, and W. Philips, “Combined wavelet-domain and motion-compensated video denoising based on video codec motion estimation methods,” *IEEE transactions on circuits and systems for video technology*, vol. 19, no. 3, pp. 417–421, 2009.

[9] F. Luisier, T. Blu, and M. Unser, “Sure-let for orthonormal wavelet-domain video denoising,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 913–919, 2010.

[10] J. C. Brailean, R. P. Kleihorst, S. Efstratiadis, A. K. Katsaggelos, and R. L. Lagendijk, “Noise reduction filters for dynamic image sequences: A review,” *Proceedings of the IEEE*, vol. 83, no. 9, pp. 1272–1292, 1995.

[11] V. Zlokolica, A. Pizurica, and W. Philips, “Noise estimation for video processing based on spatio-temporal gradients,” *IEEE Signal Processing Letters*, vol. 13, no. 6, pp. 337–340, 2006.

[12] V. Kamble and K. Bhurchandi, “Noise estimation and quality assessment of gaussian noise corrupted images,” *IOP Conference Series: Materials Science and Engineering*, vol. 331, no. 1, pp. 1–10, 2018.

[13] M. Ghazal, A. Amer, and A. Ghrayeb, “A real-time technique for spatio–temporal video noise estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 12, pp. 1690–1699, 2007.

[14] S.-M. Yang and S.-C. Tai, “A fast and reliable algorithm for video noise estimation based on spatio-temporal sobel gradients,” in *International Conference on Electrical, Control and Computer Engineering 2011 (InECCE)*. IEEE, 2011, pp. 191–195.

[15] K. Miyamae and S. Gohshi, “Noise level detection in general video,” in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Jan 2018, pp. 1–4.

[16] “Ite/arib hi-vision test sequence 2nd edition,” 2009.

[17] M. J. B. Wilhelm Burger, *Principles of Digital Image Processing: Fundamental Techniques*. Springer, 2009, pp. 144–145.

