Temporal noise reduction for preprocessing of video streams in monitoring systems

Olgierd Stankiewicz
Institute of Electronics and Telecommunications
Division of Multimedia Telecommunications and Radioelectronics,
Poznan University of Technology,
Piotrowo 3A, 60-695 Poznan, Poland,
e-mail: ostank@multimedia.edu.pl

Adam Łuczak
Institute of Electronics and Telecommunications
Division of Multimedia Telecommunications and Radioelectronics,
Poznan University of Technology,
Piotrowo 3A, 60-695 Poznan, Poland,
e-mail: aluczak@multimedia.edu.pl

Antoni Roszak
Institute of Electronics and Telecommunications
Division of Multimedia Telecommunications and Radioelectronics,
Poznan University of Technology,
Piotrowo 3A, 60-695 Poznan, Poland,
e-mail: aroszak@et.put.poznan.pl

Abstract — **This paper presents an efficient preprocessing algorithm for dealing with noise in video sequences. The algorithm is based on simple thresholding and linear filtering. First stage of proposed technique is motion detection which separates steady regions from moving ones. Second stage involves temporal filtering of steady regions. That yields with "freezed" blocks that are more susceptible for compression. The proposed algorithm is appropriate for preprocessing stage before compression of video sequences.**

I. INTRODUCTION

Progress in modern compression techniques comes at expense of complexity and computational power requirements. Typical gain that is introduced by new improvements, oscillate around few percent of compressed stream rate, while consuming considerable amount of memory and processing time. It is worth to notice, that often more can be attained if fidelity principle is abandoned and replaced with expectation of perceptual quality. Among techniques where fidelity deterioration is not related to perceptual degradation, the most useful are noise reduction methods, because they don't introduce loss of information. There are several techniques that reduce noise. They can be divided into three groups:

- spatial filtering – bases on detection of noise in single frame domain; its main weakness is that it introduces blur and degradation of detail
- temporal filtering – model of the noise is constructed for each pixel independently as a random process in time; it doesn't work well on sequences with fast motion
- spatio-temporal filtering – noise reduction exploits information from neighbor pixels both in space and time domain; this method incorporates advantages of two former, but requires more complex computations.

Region of interest for this paper includes monitoring. Because detail preservation is important in this application, spatial filtering is deprecated. Due to complexity spatio-temporal filtering is also not possible. Authors propose to use simple temporal filtering in regions where it applies best – to steady regions of the sequence. Because motion detection phase is not critical for efficiency of whole process, it is appropriate to use least expensive suitable solution. It has turned out that thresholding of differences between pixels of consecutive frames done in blocks is satisfactory.

II. ALGORITHM

Proposed algorithm consists of two main phases: motion detection and noise filtering. Both phases are done in rectangular blocks. Size of single luminance block is a parameter to presented algorithm and varies from 2x2 to 16x16 pixels. Size of chrominance blocks corresponds to same spatial size as luminance, thus for 4:2:0 sampling scheme they vary from 1x1 to 8x8.

At first, each block is classified as moving or steady, with respect to differences between consecutive frames and processed reference frame. Differences that exceed given thresholds classify blocks as moving ones. Those blocks are left unchanged during processing. Blocks classified as steady are linearly filtered with respect to previous frames. The idea behind this is to "freeze" the noise so that motion estimation in hybrid encoder is not confused with fast varying noise. From encoder's point of view, it would be best to entirely discard motion in blocks classified as steady, but unfortunately it introduces unpleasant artifacts when contents of block start to move.

III. MOTION DETECTION

As noted before, the whole processing is done in rectangular blocks. Two modes of operation are proposed for this phase, which involve one or two thresholds.

First one is simpler to implement, requires less processing power (thus is faster), but as our results presented in fig. 1 show, is also much less effective.

The latter method is more complex but offers significant improvement both in PSNR and bitrate. Both methods are virtually identical from perceptual point of view.

In both schemes, for each block, composed of absolute pixel differences $d(x,y)$ between currently processed reference frame and consecutive frames,

$$d(x,y) = \left| f(x,y) - r(x,y) \right| \quad (1)$$

where: $d(x,y)$ - absolute difference block

$f(x,y)$ - currently processed frame

$r(x,y)$ - reference frame

a maximum absolute difference value M is calculated:

$$M = \max\left[d(x,y)\right] \quad (2)$$

In case of using two thresholds, number of absolute differences N that exceed given value, is also taken into account:

$$N = \frac{count\ of\ \left[d(x,y) > d_{treshold}\right]}{B} \quad (3)$$

where B stands for area occupied by single block (in square pixels).

If any of conditions:

$$M > M_{motion} \quad \text{or} \quad N > N_{motion}, \quad (4)$$

is true, analyzed block is classified as moving, and is excluded from further processing.

All threshold parameters: $d_{treshold}$, M_{motion} and N_{motion} were found experimentally. Because design goal for the algorithm was not to detect motion as an objective characteristic of the signal, but mainly in perceptual manner, it can be noticed that those thresholds do not depend on input signal but rely on human perceptual model.

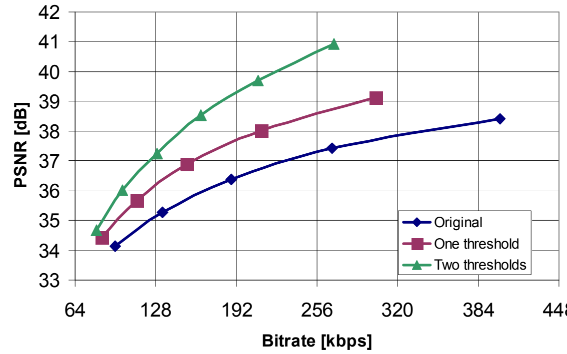

Fig. 1. Motion detection mode comparison.

Because of superiority of “two thresholds” method over “one threshold”, it was used in all remaining experiments that were made and that are presented in further sections.

IV. THRESHOLD ADAPTATION

Although it would be optimal to calculate the motion estimation thresholds for each sequence separately, it would be necessary to assume that algorithm has some primal knowledge about sequence quality. Since it is difficult to distinguish noise from subtle movement and nearly impossible to do so in case of chaotic movement (e.g. waves on water), authors assume that presented de-noising algorithm requires feedback from user. Thus, best result can be achieved with manual setting of motion estimation thresholds. In considered region of interest (monitoring systems) it is not a big disadvantage since threshold calibration needs to be done only once for each stream source.

V. NOISE FILTERING

Blocks classified as steady are filtered in time domain in order to suppress noise. In perfect case, transmittance of filter should correspond to shape of PSD (power spectral density) of the additive noise, but unfortunately exact noise model of the noise is not known a priori. Since our primary goal is not to discard noise entirely, but only assure that it won’t mislead motion compensation module, we decided to find best filter characteristic empirically.

Filtering is the most time-consuming operation in our algorithm and we decided to incorporate IIR filter in our design. To minimize non-linearity of phase that would otherwise introduce distortions we chose low-order Butterworth type filters. Some examples are presented below in fig. 2-4.

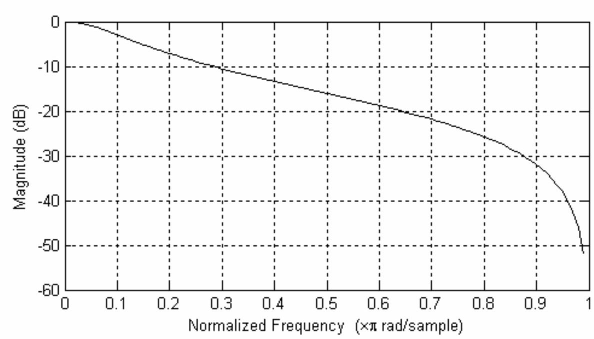

Fig. 2. First order Butterworth filter w = 0.1.

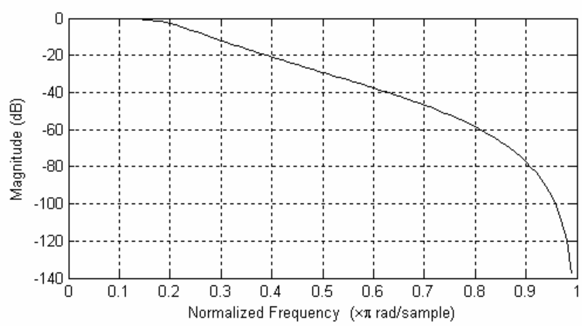

Fig. 3. Second order Butterworth filter w = 0.2.

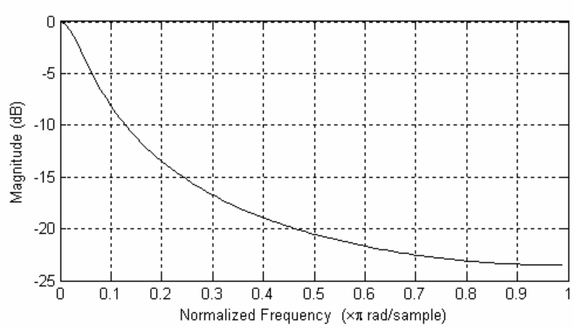

Fig. 4. First order Butterworth filter w = 0.1, with zeroed higher nominator coefficients (except of one). Chosen for further evaluation as an effect of excellent subjective performance.

VI. RESULTS

Results are presented for two sequences: "Clock" sequence and "Hall monitor" sequence. In authors opinion, those two sequences cover overall field of applications very well.

"Clock" sequence is just a wall clock ticking; the whole scene is contaminated by noise. Ideally, the background would be coded only once and the following frames should carry no background texture information at all.

"Hall monitor" sequence is widely known and often used in various presentations concerning MPEG coding algorithms. High magnitude noise (both in luminance and chrominance) existing in this sequence is probably a result of using low-end analog recording device.

Fig. 5. Frame from original "Clock" sequence.

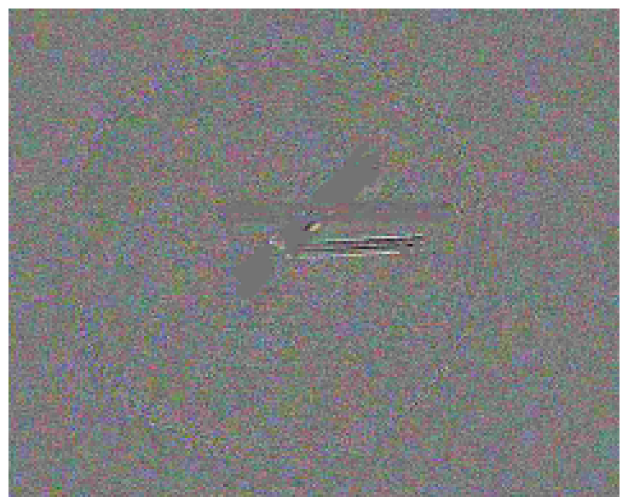

Fig. 6. Difference between original "Clock" sequence and filtered with presented algorithm (with contrast enhancement).

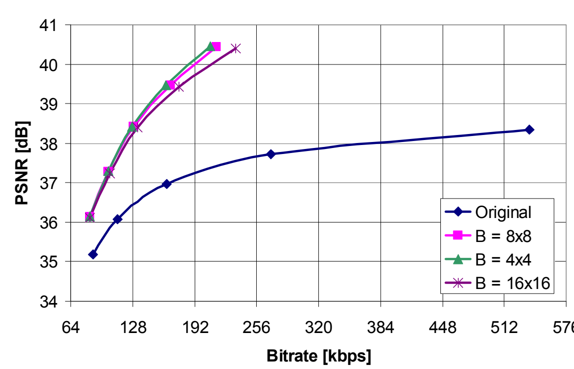

Fig. 7. Performance of presented algorithm on "Clock" sequence.

Various other sequences with different characteristics were tested and results resemble those attained on "Hall monitor", and lead to similar conclusions.

Fig. 8. Frame from original "Hall monitor" sequence.

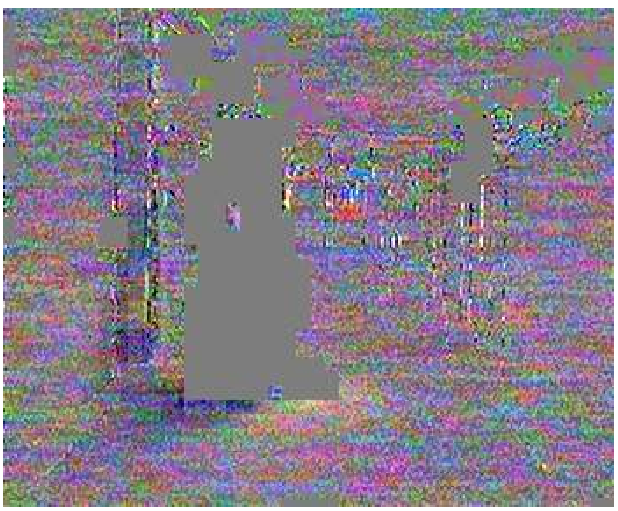

Fig. 9. Difference between original "Hall monitor" sequence and filtered with presented algorithm (with contrast enhancement).

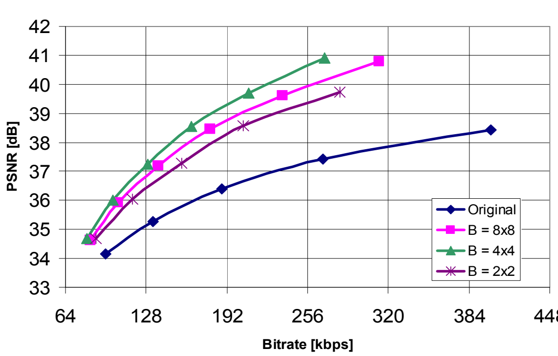

Fig. 10. Performance of presented algorithm on "Hall monitor" sequence.

Please notice that only moving regions of the scene are passed through algorithm without modification (grey blank fields – fig. 6 and fig. 9). Remaining regions are filtered temporally, which practically removes noise from the whole scene.

When taking performance results above into consideration (Fig. 7 and Fig. 10), it should be noticed that PSNR was acquired in compression stage. It is not the same as comparing to hypothetical noiseless sequence. Since such sequence is not likely to be available (because there would be no point in processing the noisy version then), evaluation must depend on subjective examination. Our research shows that most of the spectators are likely to believe that output sequence of devised algorithm is, in fact, the original one. Even if it was not the case, our results prove that fidelity of compressed sequence is improved. The best results were achieved for 4x4 block size, since H.264/AVC reference codec was used for testing.

VII. CONCLUSION

An efficient scheme of noise suppression technique utilizing selective temporal filtering was presented in this paper. The results are promising and show that for the same bitrate, coding quality measured with respect to PSNR is much higher (up to 3dB).

Presented algorithm can be successfully used in video surveillance systems as preprocessing stage. Number of bits used for background representation is drastically reduced. That yields in fact, that noise and distortions, introduced by lossy coding, are distributed over steady regions, saving bandwidth for more accurate representation of actual scene movement. While maintaining low bitrate, algorithm allows for preservation of much more detail in the compression stage, than might have been discarded during filtering.

Devised technique brings significant improvement in matter of bitrate for a low computational cost and almost no quality penalty, which makes is competitive for advanced control algorithms.

REFERENCES

[1] K. Arakawa, "Median filter based on fuzzy rules and its application to image restoration," *Fuzzy Sets Syst.*, 1996.

[2] M. Lang, H. Guo, J.E. Odegard.; C.S. Burrus,, R.O. Wells "Noise reduction using an undecimated discrete wavelet transform" Signal Processing Letters, IEEE Volume 3, Issue 1, Jan. 1996

[3] P. R. Schrater, D. C. Knill and E. P. Simoncelli "Mechanisms of visual motion Detection" Nature America 2000

[4] T. Wittkop, V. Hohmann, and B. Kollmeier, "Noise reduction strategies employing interaural parameters," *J. Acoust. Soc. Amer.*, vol. 105, no. 2, 1999

[5] D. Kersten, Statistical efficiency for the detection of visual noise. Vision Res. 27, 1029–1043 (1987).