Video denoising using multiple class averaging with Multiresolution

Vladimir Zlokolica, Aleksandra Pizurica, and Wilfried Philips

IPI, TELIN, University of Ghent, Sint-Pietersnieuwstraat 41,
9000 Ghent, Belgium
{vzlokoli, sanja, philips}@telin.ugent.be
http://telin.ugent.be/~vzlokoli

Abstract. This paper presents a non-linear technique for noise reduction in video that is suitable for real-time processing. The proposed algorithm automatically adapts to detected levels of detail and motion, but also to the noise level, provided it is short-tail noise, such as Gaussian noise. It uses a one-level wavelet decomposition, and performs independent processing in four different bands in the wavelet domain. The non-decimated transform is used because it leads to better results for image/video denoising than the decimated transform. The results show that from both a PSNR and a visual quality, the proposed filter outperforms the other state of the art filters for different image sequences.

1 Introduction

Video sequences are often corrupted by noise, e.g., due to bad reception of television pictures. Some noise sources are located in a camera and become active during image acquisition under bad lightning conditions. Other noise sources are due to transmission over analogue channels. In most cases the noise is white and gaussian, and in some cases low-level impulse noise (which we do not consider in this paper).

Noise reduction in image sequences is used for various purposes, e.g. for visual improvement in video surveillance. It is achieved through some form of linear or non-linear operation on correlated picture elements. In the recent past a number of non-linear techniques for video processing have been proposed [1–4] and were proved superior to linear techniques.

Video denoising is usually done by temporal-only [5, 6] or spatio-temporal [7, 8, 4] filtering. The third possibility (spatial-only filtering) is rarely considered in the literature, perhaps because it often leads to quite visible artifacts. It is generally agreed that in the case of low noise corruption, which is important in many real video applications, spatio-temporal filtering performs better than temporal filtering [7]. However in the case of spatio-temporal filtering there is a danger of significantly reducing the effective resolution of video, i.e. spatial blurring, especially in case of spatio-temporal recursive filtering. In general, the best performance can be achieved by exploiting information from both future and past frames, but this leads to a delay of at least one frame which is undesirable in some real-time applications. For this reason, many algorithms exploit information from past frames only (usually the current frame and one or two previous frames).

In any case dealing correctly with motion is a very important issue in video processing. There are two general approaches for dealing with motion:

- Motion estimation and compensation [5, 9]
- Motion detection and performing some special operations in case of detected motion [1, 4]

Examples of the first case, are techniques that apply a time-recursive filter over an estimated motion trajectory. This approach yields good results provided the motion estimation is accurate. In practice for computational reasons the motion estimates are not accurate enough, which can cause certain artifacts.

In the second approach, based on the output of the motion detector, a spatio-temporal filter is tuned to avoid motion blur in case of motion, and to filter as much as possible in case of no motion. Since the motion detection is imprecise due to noise, the filter must find a compromise between noise reduction and blurring.

In this paper, we propose an algorithm that allows fast, real-time implementation. It is based on spatio-temporal recursive filtering and multiple threshold averaging. It automatically adapts to motion - reducing the contribution of the pixels in the previous fields, and to detail. We explain the main principle in section 2 and extend it to the wavelet domain in section 3. In section 4 we present experimental results and a comparison with other techniques. Finally in section 5 we present conclusions and give possible directions for further research.

2 Adaptive multiple class averaging in the base domain

In this paper we present a spatio-temporal recursive filter, based on multiple threshold filtering. The idea was inspired by the *still image processing* technique [10] where a sigma filter was proposed. The sigma filter takes all pixel values within a "current" 3×3 window for which the absolute difference to the central pixel value is less than or equal to two times the standard deviation σ of Gaussian noise and averages them to produce an output. The idea is that 95% of random samples lie within the range of two standard deviations. Any pixel outside the 2σ range most likely comes from a different population (e.g. on the other side of an edge) and, therefore should be excluded from the average. Due to the binary weighting coefficients (zero for pixel values for which the absolute difference to the central pixel value is higher than 2σ, and one for the other pixel values) a shot-noise like effect occurred in the processed images, which is very annoying to the viewer. This happened because for higher noise values there were not enough or any pixel values in the neighborhood with weighting coefficient one, and thus they remained unfiltered.

In order to avoid shot-noise like artifacts, our method proposes the following. We classify grey pixel values into four different classes and weight them according

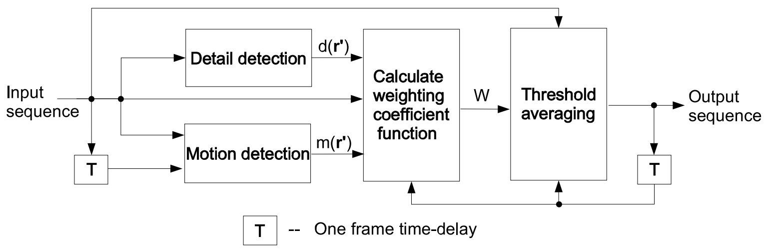

Fig. 1. The General filter description in the base domain

to their class index in the averaging process, where classes are defined according to the absolute difference between the pixel value and the central pixel value.

In this section we propose the general description of the filter method in the base (non-transform) domain. In the following we denote an image pixel as $I(x, y, t)$, where (x, y) and t indicate the spatial and temporal location, respectively. We consider a $3 \times 3 \times 2$ sliding window, that consists of 3×3 pixels in the current and previous frame, with $\mathbf{r} = (x, y, t)$ being the central pixel position, in the current frame, and $\mathbf{r}' = (x', y', t')$ being any pixel position in the sliding window. In the remainder of the paper we will use terms 'current window - CW' and 'previous window - PW' that correspond to the pixel values of the $3 \times 3 \times 2$ sliding window from the current and the previous frame respectively.

The general description of the algorithm is shown in Fig.1. There, the function of spatial detail, $d(\mathbf{r})$, equals the local dispersion of the current window. In the same figure, $m(\mathbf{r})$ is a measure for the amount of detected motion and is defined as the difference between the average grey value of the current window and the average grey value of the previous window.

We mathematically define the output of our new filter $O(\mathbf{r})$ as follows:

$$O(\mathbf{r}) = \frac{\sum_{\mathbf{r}'} W(i(\mathbf{r}', \mathbf{r}), d(\mathbf{r}), m(\mathbf{r}), t') I(\mathbf{r}')}{\sum_{\mathbf{r}'} W(i(\mathbf{r}', \mathbf{r}), d(\mathbf{r}), m(\mathbf{r}), t')}, \tag{1}$$

where the weights $W(i(\mathbf{r}', \mathbf{r}), d(\mathbf{r}), m(\mathbf{r}), t')$ in (1) for a particular pixel $\mathbf{r}'$ in the window depend on the amount of detail $d(\mathbf{r})$ and motion $m(\mathbf{r})$ in the current window. Furthermore, they depend on the difference in grey scale $|I(\mathbf{r}') - I(\mathbf{r})|$ through the class index $i(\mathbf{r}', \mathbf{r})$ which can assume 4 values, and on weather $\mathbf{r}'$ is in the current frame or in the previous frame, $t'= 0$ or $t'= 1$, respectively. The lowest index value $i = 0$ corresponds to pixel values that are closest to the central pixel value. Taking that into account, we intend to give more importance to lower index classes in case of big spatial detail, to avoid blurring. On the other hand in case of small spatial detail we intend to give similar importance to all classes in order to perform stronger smoothing.

Although noise will be less reduced in case of bigger spatial detail, this is not a problem: such regions contain high spatial frequencies and according to [9, 11] the human eye is not very sensitive to those frequencies any way.

Table 1. Values of constants

Constants	non-transform domain	wavelet domain
K_1	0.1105	0.1105
K_2	0.00102	0.00305
K_3	0.0669	0.0669
K_4	0.0142	0.0284
k	1.5	0.5

For each pixel $\mathbf{r}'$, we define the absolute difference with the central pixel, $\Delta(\mathbf{r}', \mathbf{r})$, as follows:

$$\Delta(\mathbf{r}', \mathbf{r}) = |I(\mathbf{r}') - I(\mathbf{r})|, \tag{2}$$

according to which, four different classes $i(\mathbf{r}', \mathbf{r})$ can be distinguished, in the following way:

$$i(\mathbf{r}', \mathbf{r}) = \begin{cases} 0,\ \Delta(\mathbf{r}, \mathbf{r}') \leq k\sigma_n \\ 1,\ k\sigma_n < \Delta(\mathbf{r}', \mathbf{r}) \leq 2k\sigma_n \\ 2,\ 2k\sigma_n < \Delta(\mathbf{r}', \mathbf{r}) \leq 3k\sigma_n \\ 3,\ \Delta(\mathbf{r}', \mathbf{r}) > 3k\sigma_n \end{cases} \tag{3}$$

The optimal values for the thresholds used for distinguishing classes were found experimentally, and the value of k is given in Table 1. For each pixel, $\mathbf{r}'$, that belongs to a certain class i the weighting function $W(i, d, m, t')$ is assigned in the following way:

$$W(i, d, m, t') = \begin{cases} \exp(-i/(\eta(d)\sigma_n))\beta(m, t'), & i = 0, 1, 2 \\ 0, & i = 3 \end{cases} \tag{4}$$

where $\eta(d)$ is used to modify the exponential function in (4) depending on the locally measured spatial detail in image, d. In addition, σ_n represents the standard deviation of Gaussian noise estimated in the video.

We have experimentally found an appropriate shape of the function $\eta(d)$, as follows:

$$\eta(d) = K_1 \exp(-K_2\, d) + K_3 \exp(-K_4\, d), \tag{5}$$

where the values of the constants $K_j, j = 1, \ldots, 4$ are shown in Table 1. The main idea behind this function is that it is inversely proportional to d, that is in case of bigger spatial detail it should produce lower values, and vice versa. This way $\eta(d)$ will influence the slope of the exponential function in (4), in order to give more importance to lower class indices i in case of bigger spatial detail. However, the performance of the filter also depends on the shape of $\eta(d)$. The particular choice in (5) works well but more research is needed to find the best choice.

The function $\beta(m, t')$ in (4) is meant to make the filter more robust against motion. This function limits the contribution of the pixels from the previous window in case of motion. Pixel values from the previous window yield a smaller contribution than otherwise similar pixels from the current window. The bigger m the smaller the contribution of the pixels from the previous window is. On the contrary to other algorithms which use binary logic (motion: no - motion: yes) we introduced fuzzy logic in our motion detection. The fuzzy logic is introduced through the function $\beta(m, t')$ that takes values in range the $[0, 1]$ and is defined as follows:

$$\beta(m, t') = \begin{cases} 1, & t' = 0 \\ \exp(-\gamma m), & t' = 1 \end{cases}, \tag{6}$$

where the parameter γ is used to control the sensitivity of the motion detector, i.e. the shape of the function $\beta(m, t')$. The greater γ the more sensitive the motion detector will be, and the greater the contribution of the pixel values from the previous frame will be in the final output of the filter. The value of γ was experimentally determined in order to get the best PSNR, for four test sequences. We found that $\gamma = 1/(2\sigma)$ is the optimal value.

3 Adaptive multiple class averaging in the wavelet domain

The wavelet transform [12] naturally facilitates spatially adaptive algorithms. It compresses essential information in an image into relatively few large coefficients, that correspond to the main image details at different resolution scales.

In our application we have used a non-decimated transform with the quadratic spline-wavelet [12, 13]. We have used only one level in the decomposition for the sake of simplicity and time cost.

The general description of the algorithm is given in Fig. 2. First the direct wavelet transform is performed and four different bands LL, LH, HL and HH are obtained. After that, the LL, LH, HL and HH bands are processed with filters that are special cases of the filter of section 2, which are specifically tuned to the properties of each of the subbands. We call these filters LLF, LHF, HLF and HHF filters, respectively.

The LLF filter is a simplified version of the filter described in section 2. The function $W(i, d, m, t')$ in (4) now depends only on the class index i, i.e. the pixel grey value, and is defined as follows:

$$W(i, d, m, t') = \begin{cases} 1, & i = 0 \\ 0.2, & i = 1 \\ 0.1, & i = 2 \\ 0, & i = 3 \end{cases} \tag{7}$$

where the border values used for multiple thresholding are adapted to the wavelet domain, using an appropriate value of k in the (3), which is shown in Table 1.

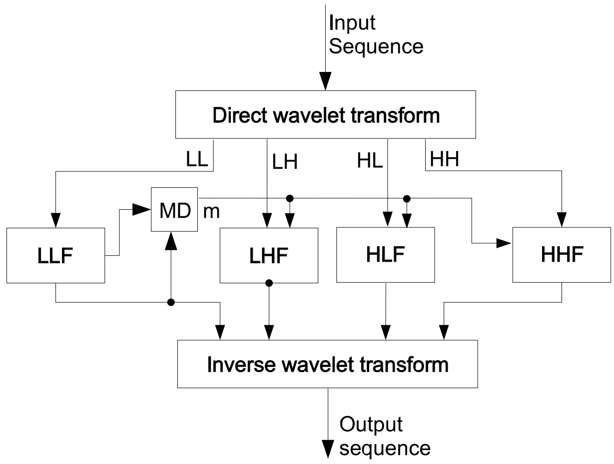

Fig. 2. The General filter description for the wavelet based filtering

The Filters LHF, HLF and HHF are the basically equal to the filter described in section 2 except for the following changes:

• The value of k in (3) is adapted to the wavelet domain, and is shown in Table 1.

• The constant values $K_i, i = 1, \ldots, 4$ were experimentally tuned to optimize the performance of the filter, i.e. to adapt to the wavelet domain, and are presented in Table 1.

• The motion parameter m is no longer computed internally (in LHF, HLF, HHF) but is now computed on the filtered LL band, i.e. on the output of the LLF filter.

After all four bands HH, HL, LH and LL have been processed, an inverse wavelet transform is done, which produces the output sequence.

4 Experimental results

To evaluate the results of the proposed time-recursive filter in the base and the wavelet domain, in the presence of white Gaussian noise, both peak signal to noise ratio (PSNR) and visual evaluation were used. The PSNR values equally high and low frequency components, whereas the human eye is less sensitive to high frequency components. Thus, both the PSNR and visual evaluation were taken into account to give the final evaluation of the result.

The results are compared with those of the state of the art rational filter [4] ('Rational'), the 3D K-NN filter [14] ('3D K-NN'), and the adaptive 3D K-NN filter [15] ('Adaptive K-NN'). In Fig. 3, the filters are compared in terms of PSNR, for the 'Salesman', the 'Flower Garden', the 'Trevor' and the 'Miss

America' sequences respectively, for the case of Gaussian noise, $\sigma = 10$. It should be noted that all test sequences were grey-scale images with pixel value 0 - 255.

In addition the notations 'WAVTHR' and 'THRF' in the PSNR graphs stand for the filter explained in section 3 and section 2, respectively.

The visual evaluation has determined that our method performs much better in comparison to the other mentioned methods. The original and processed sequences can be found on the web: http://telin.ugent.be/˜vzlokoli/VLBV03/. It preserves image details well and at the same time sufficiently clears the noise in non-detailed parts of the image. The PSNR obtained by the proposed 'WAVTHRF' filter is not only bigger on average for each of the test sequences, but almost on any frame. It can be seen that PSNR is fluctuating very little through the frames, and the averaged PSNR through the frames is around 1dB better than for the other methods. However we realize that PSNR is not always a good indication of the visual quality, so we also judge the visual quality. From this point of view the proposed method proved superior on all four sequences.

5 Conclusion

A time recursive spatio-temporal filter has been presented in this paper. It is consistently better than other methods and relatively simple. Although, the computation time is relatively high the algorithm could be adapted for real-time implementation, e.g. by piece-wise linear approximation of functions (4-6) or quantizing certain parameter values in the algorithm, without a big loss of performance. Further research could be aimed at using different wavelet functions for decomposition, decimated transform which could demand less computation time, or using more wavelet decomposition levels. In addition, improved motion detection, or motion estimation could be included in the algorithm.

References

1. K. Jostschulte, A. Amer, M. Schu, and H. Schroder, "Perception adaptive temporal tv-noise reduction using contour preserving prefilter techniques," *IEEE Trans. on Consumer Electronics*, vol. 44, no. 3, pp. 1091–1096, 1998.
2. G. De Haan, T.G. Kwaaitaal-Spassova, M.M. Larragy, O.A. Ojo, and R.J. Schutten, "Television noise reduction ic," *IEEE Trans. on Consumer Electronics*, vol. 44, no. 1, pp. 143–153, 1998.
3. G. De Haan, T.G. Kwaaitaal-Spassova, M.M. Larragy, and O.A. Ojo, "Memory integrated noise reduction ic for television," *IEEE Trans. on Consumer Electronics*, vol. 42, no. 2, pp. 175–180, 1996.
4. F. Cocchia, S. Carrato, and G. Ramponi, "Design and real-time implementation of a 3-D rational filter for edge preserving smoothing," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 4, pp. 1291–1300, Nov. 1997.
5. Gerhard de Haan, "Ic for motion-compensated de-interlacing, noise reduction and picture rate conversion," *IEEE Trans. on Cosumers Electronics*, vol. 45, no. 3, pp. 617–623, Aug. 1999.

6. Rajesh Rajagopalan, "Synthesizing processed video by filtering temporal relationships," *IEEE Trans. on Image Processing*, vol. 11, no. 1, pp. 26–36, Jan. 2002.
7. K.O. Mehmet, S. Ibrahim, and T. Murat, "Adaptive motion-compensated filtering of noisy image sequences," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 4, pp. 277–290, Aug. 1993.
8. D. Dugad and N. Ahua, "Noise reduction in video by joint spatial and temporal processing," *Submitted for review for IEEE Trans. CVST,2001.*
9. E.B. Bellers and G. De Haan, *De-Interlacing:A Key Technology for Scan Rate Conversion*, Elsevier Science B.V., Sara Burgerhartstraat, Amsterdam, 2000.
10. Jong-Sen Lee, "Digital smoothing and the sigma filter," *Computer Vision Graphics and Image Processing*, vol. 24, pp. 255–269, 1983.
11. "Ccir recommendation 421-1, annex iii," 1967.
12. Stephane Mallat, *A wavelet tour of signal processing (2nd ed.)*, Academic Press, Oval Road, London, 1999.
13. Stephane Mallat and Sifen Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, July 1992.
14. V. Zlokolica, W. Philips, and D. Van De Ville, "A new non-linear filter for video processing," in *IEEE Benelux Signal Processing Symposium*, Mar. 2002, pp. 221–224.
15. V. Zlokolica and W. Philips, "Motion- detail adaptive k-nn filter video denoising," in *Report 2002, http://telin.ugent.be/˜vzlokoli/Report2002vz.pdf.*

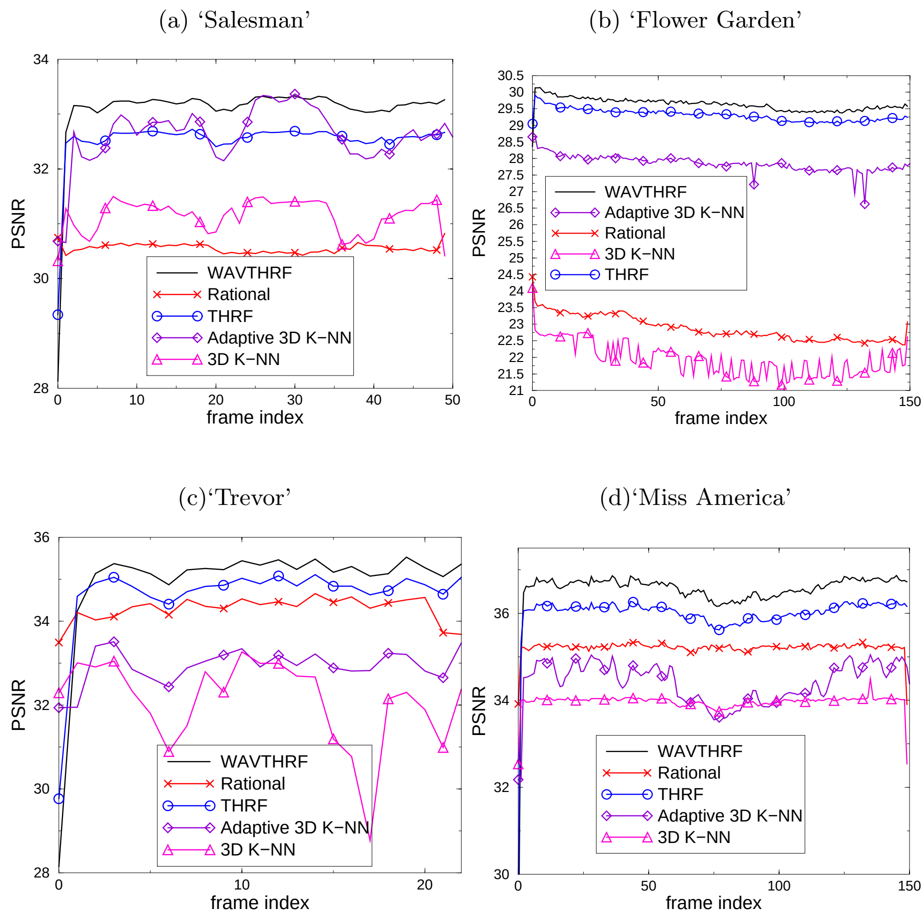

Fig. 3. Comparison in terms of PSNR for different sequences