RECURSIVE TEMPORAL DENOISING AND MOTION ESTIMATION OF VIDEO

V. Zlokolica, A. Pižurica, W. Philips

Ghent University, Dept. of Telecommunications and Information Processing (TELIN), IPI
Sint-Pietersnieustraat 41, 9000 Gent, Belgium
Vladimir.Zlokolica@telin.UGent.be

ABSTRACT

In this paper we present a new technique for video denoising which is based on a novel motion estimation algorithm. First, recursive temporal denoising is performed through the estimated motion trajectory. After that appropriate spatial filtering is done. The proposed algorithm automatically adapts to the detected noise level, provided it is short-tail noise, such as Gaussian noise. It uses a one-level wavelet decomposition where both motion estimation and denoising is performed.

The non-decimated transform is used because it is nearly shift invariant and thus yields better motion estimation and denoising results than the decimated transform. The results on different image sequences demonstrate that the proposed filter outperforms the other state-of-the-art filters both in terms of PSNR and visually.

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 not only improves the visual quality but also increases the performance of subsequent image processing tasks, such as coding, analysis, or interpretation. 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, 2] and were proved superior to linear techniques. Video denoising is usually done by temporal-only [3, 4] or spatio-temporal [1, 2] filtering. Methods that attempt at fully exploiting a great temporal redundancy of video apply so-called $motion - compensated$ filtering, i.e., filtering through estimated motion trajectory [3, 5]. A thorough review of noise reduction algorithms for digital image sequences is presented in [5].

It is generally agreed that spatio-temporal filtering performs better than temporal filtering [2, 6]. However spatio-temporal filtering poses a threat of significantly reducing the effective resolution of video (spatial blurring), especially in case of spatio-temporal recursive filtering or in the case when spatial filtering precedes the temporal one. On the other hand temporal denoising although preserving the full resolution of the input image sequence can produce disturbing artifacts due to the imperfections of motion estimates. Moreover, in the case of recursive temporal filtering these errors will propagate through the sequence. To overcome the latter problem one must reduce filtering when no accurate motion vectors are found. [6, 7].

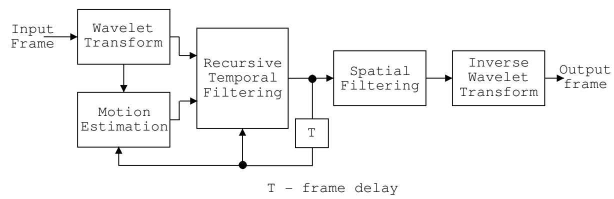

Fig. 1. The general framework description of the proposed algorithm

In this paper, we propose a new wavelet domain video denoising scheme which consists of motion compensated recursive temporal recursive filtering followed by spatial denoising. In contrast to some existing sequential spatio-temporal schemes [8] where the spatial filter precedes the temporal one, we apply temporal filtering to the noisy sequence first such as in [2] and then subsequently use the spatial filter (see Fig.1). By temporal filtering through the motion estimated trajectory we efficiently exploit temporal redundancy in order to suppress the noise, $without\ spatial\ blurring$. If a reliable motion vector cannot be found we reduce the amount of temporal recursive filtering proportionally so the errors do not propagate through the processed sequence. By subsequent adaptive spatial filtering we remove the remainder of noise. As a result, by not employing the spatial filter in the recursive temporal scheme we reduce the spatial blur in the denoised image sequences. We use a non-decimated transform with the quadratic spline-wavelet [9, 10]. Our current implementation uses only one-level wavelet decomposition. The main novelty of the proposed method is an original noise robust motion estimation scheme in the wavelet domain.

The paper is organized as follows. We present the proposed method in Section 2. In subsections 2.1 and 2.2 we describe the new algorithm for motion estimation and temporal recursive filtering, respectively, and in Section 2.3 we explain spatial filtering. In Section 3 we present experimental results and we conclude the paper in Section 4.

2. THE PROPOSED TECHNIQUE

The general description of the proposed algorithm for video processing is presented in Fig. 1. First, one level wavelet transform is performed on the input current frame (ICF) and four different bands LL_c, LH_c, HL_c and HH_c are obtained. They are used for

0-7803-8554-3/04/$20.00 ©2004 IEEE.

motion estimation together with previously processed bands LL_p, LH_p, HL_p and HH_p for the preceding frame, i.e. for determining motion vectors (MV) for the current frame in the correspondence to the previously processed frame (PPF).

Next, the estimated MVs are used for recursive temporal filtering (RTF) in the wavelet domain in all four bands. The estimated accuracy of the MVs is used to control the amount of filtering: we filter less in those areas where estimated MV are not reliable enough and more in the opposite case. The current output wavelet bands of the RTF: LH_o, HL_o and HH_o are further processed by a spatial filter. The spatial filter suppresses the remaining noise left after temporal recursive filter. It especially affects the areas where sufficient temporal filtering could not be performed due to not reliable enough motion estimates. Finally, the inverse wavelet transform is applied to the spatially processed bands resulting in the output frame.

2.1. Motion Estimation

Motion estimation in the wavelet transform domain has received considerable attention in the last few years. However motion estimation in the wavelet domain (in case of decimated transform) is highly dependent on the alignment of the signal and the discrete grid chosen for analysis [12]. Due to shift-variance, motion estimation and compensation of the wavelet coefficients is difficult. One solution to overcome the latter problem was presented in [12].

Another solution is to use the non-decimated wavelet transform (WT) which ensures approximate shift-invariance. The non-decimated WT removes the downsampling operation from the traditional critically sampled decimated WT to produce an overcomplete representation [11]. The size of each subband LL, LH, HL and HH is exactly the same as that of the input signal. An example of recursive motion estimation and compensation in a non-decimated wavelet transform was presented in [13].

In the mentioned methods for motion estimation and compensation in the wavelet domain [11–13] the motion estimation is generally based on simply minimizing mean absolute difference (MAD) between the current wavelet block and the reference wavelet block. In our approach we aim at minimizing MAD separately for horizontal and vertical motion vector components and introduce penalties in the cost functions which are dependent on the estimated accuracy of the initial motion estimates. In such way we make our motion estimation scheme more robust to noise.

In our work we use LL, LH, HL bands to perform motion estimation. The LL band corresponds to the lowest frequencies, while the LH and HL corresponds to vertical and horizontal high frequencies (spatial edges), respectively. The highest frequency subband HH (corresponds to diagonal edges) was not used because in the one-level WT it contains a significant amount of noise. We apply a block matching technique (in the wavelet domain) with a separable 3-step search approach.

In our approach the two-dimensional optimization problem is split into two separate one-dimensional optimizations for estimating the horizontal and vertical components of the motion vector for each block in a frame. The proposed motion estimation algorithm aims at exploiting the information from the MAD between a certain block of wavelet coefficients from the current and the previously processed wavelet band, WB_c and WB_p, respectively. For each wavelet band (WB) we calculate the MAD_{WB} as follows:

$$MAD_{WB}(s, \mathbf{mv}) = \frac{1}{N} \sum_{\mathbf{x} \in B_s} |WB_c[\mathbf{x}] - WB_p[\mathbf{x} - \mathbf{mv}]| \tag{1}$$

where $\mathbf{mv}$ denotes motion vector and "s" denotes the index of the block in the image. "B_s" denotes the s-th block area that consists of $N = N_x \times N_y = 8 \times 8$ coefficient values, where N_x and N_y stand for number of rows and columns in the block, respectively. In addition the search area is confined to $(4N_x) \times (4N_y)$ in our work. It should be noted that bigger search area could also be applied for processing sequences with faster and large movements. However for most sequences the proposed search area was found as a best compromise between motion accuracy and time consumption for motion vector search.

In the remainder of the paper we use terms $\mathbf{mv_c}$ and $\mathbf{mv_i}$ to define candidate and initial motion vectors, respectively. The $\mathbf{mv_c}$'s at each step of the proposed motion estimation algorithm are added to the initial vector $\mathbf{mv_i}$. In the first (initial) step block matching approach $\mathbf{mv_i}$ is put to the zero motion vector. There, the $\mathbf{mv_c}$'s have following horizontal (mv_{cx}) and vertical (mv_{cy}) component values: $mv_{cx}, mv_{cy} \in \{-8, -4, 0, 4, 8\}$. They are added to the $\mathbf{mv_i}$ and tested in order to find the best match motion vector ($\mathbf{mv_b}$). In the second step $\mathbf{mv_i}$ is set equal to the $\mathbf{mv_b}$ from the first step search and the candidate vectors have the following values: $mv_{cx}, mv_{cy} \in \{-4, -2, 0, 2, 4\}$. Finally in the third step $\mathbf{mv_i}$ is put to the $\mathbf{mv_b}$ from the second step, where the candidate vectors have the following values: $mv_{cx}, mv_{cy} \in \{-2, -1, 0, 1, 2\}$.

The best match vectors estimated for each step are determined by minimizing separately the following two cost functions:

$$\begin{aligned} cost_x = MAD_{LL}(s, \mathbf{mv_i} + \mathbf{mv_c}) + \\ MAD_{HL}(s, \mathbf{mv_i} + \mathbf{mv_c}) + P_x(s, mv_{cx}, \mathbf{mv_i}) \\ cost_y = MAD_{LL}(s, \mathbf{mv_i} + \mathbf{mv_c}) + \\ MAD_{LH}(s, \mathbf{mv_i} + \mathbf{mv_c}) + P_y(s, mv_{cy}, \mathbf{mv_i}) \end{aligned} \tag{2}$$

in order to obtain horizontal and vertical component of the $\mathbf{mv_b}$, respectively. The penalties P_x and P_y were introduced in order to robustly coordinate our search process against ambiguities (such as noise) and are defined as follows:

$$\begin{aligned} P_x &= k_n |mv_{cx}| M_{HL} \\ P_y &= k_n |mv_{cy}| M_{LH} \end{aligned} \tag{3}$$

where k_n is a normalizing coefficient which changes for each step of the proposed motion estimation algorithm. In our experiments we have used: $k_n = 0.5$ for the first step block matching, $k_n = 1$ for the second and $k_n = 2$ for the third. Further on, M_{HL} and M_{LH} correspond to the inversely normalized $MADS_{HL}$ and $MADS_{LH}$ by the standard deviation of the Gaussian noise (σ) and constant $k_m = 0.5$, i.e. they are defined as follows:

$$\begin{aligned} M_{HL} = k_m \frac{\sigma}{MAD_{HL}(s, \mathbf{mv_i})} \\ M_{LH} = k_m \frac{\sigma}{MAD_{LH}(s, \mathbf{mv_i})} \end{aligned} \tag{4}$$

S		S
	C	
T		T

T-temporal neighbor
S-spatial neighbor
C-central current block

Fig. 2. The position of spatial and temporal neighbors

Using penalties P_x and P_y as shown in (3) we intend to make our search process more robust in the following way: Using (1) and initial motion vector $\mathbf{mv_i}$, we calculate $MAD_{WB}(s, \mathbf{mv_i})$ for WB = HL, LH to determine how far we are from the optimal horizontal and vertical motion vector components, respectively. The closer we are to the optimal value (MAD value is relatively small) the more importance we give to the smaller candidate motion vectors by introducing bigger penalties for bigger motion candidates. In the opposite case (MAD value is relatively big) penalties are reduced and thus bigger changes, i.e. bigger candidate vectors $\mathbf{mv_c}$ are allowed. However, it should be noted that in the latter case penalties don't play such an important role for the proposed scheme. Finally, more advantageous penalty function could also be applied but in our algorithm the proposed penalty function (3) performed sufficiently good.

After the three step approach we obtain the best match motion vector estimate ($\mathbf{mv_b}$). In the final stage of the motion estimation algorithm we compare $\mathbf{mv_{b_c}}$ of the current block with already obtained motion estimates for the spatial and temporal block neighbors ($\mathbf{mv_{b_{st}}}$) and try to find final best motion vector estimate ($\mathbf{mv_{bf}}$) by minimizing the following cost function:

$$
\begin{aligned}
costf = MAD_{LL}(s, \mathbf{mv_c}) + MAD_{LH}(s, \mathbf{mv_c}) + \\
MAD_{HL}(s, \mathbf{mv_c}) + P_{st}
\end{aligned}
\qquad (5)
$$

where penalty $P_{st} = 2$ was taken for the spatio-temporal neighbors candidates $\mathbf{mv_c} = \mathbf{mv_{b_{st}}}$ and $P_{st} = 0$ for the $\mathbf{mv_c} = \mathbf{mv_{b_c}}$. The main intention of the last stage was to refine motion field by smoothing in spatial and temporal direction and thus obtain more coherent motion vector field. The idea was inspired by [7] where the motion recursive estimation was proposed. In our experiments we have used spatial and temporal neighbors as shown in Fig. 2.

2.2. Recursive Temporal Filtering

Recursive temporal filtering (RTF) is performed using the estimated motion vectors $\mathbf{mv_{bf}}$. However, the amount of the temporal filtering applied is of crucial importance because not only the filtered frame will be used for filtering next frame but for the motion estimation in the next frame as well. Therefore if a reliable motion vector $\mathbf{mv_{bf}}$ for a certain block could not be found we filter less proportionally. Otherwise we filter more if we consider motion estimate reliable enough.

In our work we have used a first-order temporal recursive filter. We use all four bands of the one-level wavelet transform in the current frame LL, LH, HL and HH and the previously processed frame LL_p, LH_p, HL_p and HH_p as follows:

$$
\begin{aligned}
LL[\mathbf{x}] &= \alpha LL_p[\mathbf{x} - \mathbf{mv_{bf}}] + (1-\alpha) LL[\mathbf{x}] \\
HL[\mathbf{x}] &= \alpha HL_p[\mathbf{x} - \mathbf{mv_{bf}}] + (1-\alpha) HL[\mathbf{x}] \\
LH[\mathbf{x}] &= \alpha LH_p[\mathbf{x} - \mathbf{mv_{bf}}] + (1-\alpha) LH[\mathbf{x}] \\
HH[\mathbf{x}] &= \alpha HH_p[\mathbf{x} - \mathbf{mv_{bf}}] + (1-\alpha) HH[\mathbf{x}]
\end{aligned}
\qquad (6)
$$

where the weighting factor α determines the amount of filtering. In our approach, α depends on the estimated accuracy of the current motion. Therefore, we relate α to the noise level σ and to the average displacement of the compensated frame block $A_d(s, \mathbf{mv_{bf}})$ as follows:

$$
\alpha(s, \sigma, \mathbf{mv_{bf}}) = k_{tf} \frac{\sigma}{A_d(s, \mathbf{mv_{bf}})} \qquad (7)
$$

where

$$
\begin{aligned}
A_d(s, \mathbf{mv_{bf}}) = MAD_{LL}(s, \mathbf{mv_{bf}}) + \\
MAD_{LH}(s, \mathbf{mv_{bf}}) + MAD_{HL}(s, \mathbf{mv_{bf}}) + \\
MAD_{HH}(s, \mathbf{mv_{bf}})
\end{aligned}
\qquad (8)
$$

and k_{tf} is a normalizing constant experimentally found to be $k_{tf} = 1.2$. In this way using (7) and (8) we try to estimate the accuracy, i.e. reliability of motion estimates $\mathbf{mv_{bf}}$. The smaller the $A_d(s, \mathbf{mv_{bf}})$ the higher the reliability of the motion estimated vectors $\mathbf{mv_{bf}}$ is and consequently the bigger α will be. As a result the contribution of the wavelet coefficient values from the previously processed frame will be bigger and the stronger filtering will be performed. However in areas where α is small, some noise will still be left but as soon as a reliable motion vector (e.g. in next frame) can be found, the noise will be suppressed. In this way we preserve the resolution of input sequence while suppressing the noise.

2.3. Spatial Filtering

In the proposed scheme, the spatial filter is intended to suppress the remaining noise without seriously reducing the resolution of input image sequence. The spatial filtering ($WRSF$) was performed on the one level wavelet outputs of time recursive filter: LH_o,HL_o and HH_o bands by spatially recursive threshold averaging: Only wavelet coefficient values from $2D$-(3×3)-sliding window (which involves some already spatially processed wavelet coefficients as well) for which the absolute difference with the central wavelet coefficient (of the $2D$-sliding window) value is lower than a noise-dependent threshold $THRD = \sigma/2$, are averaged. A more sophisticated spatially adaptive denoising [8] was used as well for the sake of comparison. In addition alternative [14] could have been used as well.

3. EXPERIMENTAL RESULTS

In our experiments we used several different grey-scale sequences: "Salesman", "Miss America", "Bicycle", "Trevor", "Tennis" and "The Chair". We have corrupted them with the Gaussian noise of various values of the standard deviation $\sigma = 5, 10, 15, 20$ and processed them with the proposed filter. The results of the output of the proposed temporal filter ($WRFT$) were first compared to the temporal recursive filter ($3RDS$) of [3] using recursion variable k for temporal recursive filtering as: $k(s) = k_r \sigma / MAD(s)$ where $MAD(s)$ and k_r correspond to mean absolute difference

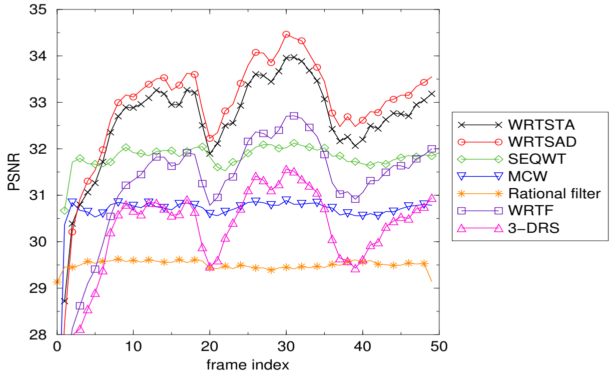

Fig. 3. The PSNR per frame for the "Salesman" sequence corrupted with the Gaussian noise ($\sigma = 15$)

Fig. 4. Up-left: noisy; Down-left: original; Up-middle: WRFT; Down-middle: 3RDS; Up-right: WRTSTA; Down-right: SEQWT

of displacement for block s for the estimated motion vectors and noise reduction parameter, respectively. In the latter the spatio-temporal neighbors used are as shown in fig. 2. After that we have tested two combinations of our temporal filter with the sequential spatial filter ($WRSF$) as explained in section 2.3 and with the spatial filter proposed in [8]. We refer to those two combinations as $WRTSTA$ and $WRTSAD$, respectively. The results were compared to other state of the art techniques for noise removal, the rational filter [1] and two wavelet based techniques: the multiclass wavelet spatio-temporal filter (MCW) of [15] and the sequential wavelet domain and temporal filtering ($SEQWT$) of [8] in terms of PSNR and visual quality of view. In comparison to [1, 15] we have found that from both visual quality of view and PSNR results are always superior. As compared to [8] PSNR performance is similar (on 50% of the sequences it is better and on 50% it is worse or the same), but visually the new filter is always better. In Fig.3 the techniques are compared using a PSNR versus frame index graph. In Fig.4 the visual result for a portion of the 29th frame of the "Bicycle" sequence corrupted with the Gaussian noise ($\sigma = 20$), is shown. Finally the visual results of the motion vector field sequences and the sequences processed by temporal only and $WRTSTA$ filter can be found on the website:http://telin.ugent.be/˜vzlokoli/icip04final.

4. CONCLUSION AND FUTURE WORK

In this paper we have proposed a new method for motion estimation and image sequence denoising in the wavelet domain. By robustly estimating motion and compensating appropriately for it we efficiently remove noise without introducing visual artifacts. In future work we intend to refine our motion estimation framework in order to deal with occlusion, and "moving block edges", i.e. refine motion vectors for blocks undergoing two or more different motions.

5. REFERENCES

[1] 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.

[2] R.P. Kleihorst, R.L. Lagendijk, and J. Biemond, "Noise reduction of image sequences using motion compensation and signal decomposition," *IEEE Trans. on Image Processing*, vol. 4, no. 3, pp. 274–284, Mar. 1995.

[3] G. 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.

[4] R. Rajagopalan and M.T. Orchard, "Synthesizing processed video by filtering temporal relationships," *IEEE Trans. on Image Processing*, vol. 11, no. 1, pp. 26–36, Jan. 2002.

[5] J. C. Brailean, R.P. Kleihorst, S. Efstratidis, K.A. Katsaggeleos, and R.L. Lagendijk, "Noise reduction filters for dynamic image sequences: A review," *Proceedings of the IEEE*, vol. 83, no. 9, pp. 1272–1292, Sept. 1995.

[6] R. Dugad and N. Ahuja, "Video denoising by combining kalman and wiener estimates," in *IEEE Conference on Image Processing*, Cobe, Japan, 1999.

[7] G. De Haan, *Video Processing for multimedia systems*, University Press Eindhoven, Eindhoven, The Nederlands, 2003.

[8] A. Pizurica, V. Zlokolica, and W. Philips, "Noise reduction in video sequences using wavelet-domain and temporal filtering," in *SPIE Conference on Wavelet Applications in Industrial Processing*, Providence, Rhode Island, USA, Oct. 2003.

[9] S. Mallat, *A wavelet tour of signal processing (2nd ed.)*, Academic Press, Oval Road, London, 1999.

[10] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, July 1992.

[11] S. Cui, Y. Wang, and E.J. Fowler, "Multihypothesis motion compensation in redundant wavelet domain," in *International Conf. on Image Processing*, Barcelona, Spain, 2003.

[12] H.W Park and H.S. Kim, "Motion estimation using low-band-shift method for wavelet-based moving-picture coding," *IEEE Trans. on Image Processing*, vol. 9, no. 4, pp. 577–587, Apr. 2000.

[13] J. Skowronski, "Pel recursive motion estimation and compensation in subbands," *Signal Processing: Image Communication*, , no. 14, pp. 389–396, 1999.

[14] S.G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. on Image processing*, vol. 9, no. 9, Sept. 2000.

[15] V. Zlokolica, A. Pizurica, and W. Philips, "Video denoising using multiple class averaging with multiresolution," in *International Workshop VLBV03, Lecture Notes in Computer Science of Springer Verlag*, Madrid, Spain, Sept. 2003.