Fuzzy logic recursive motion detection and denoising of video sequences

Vladimir Zlokolica
Aleksandra Pižurica
Wilfried Philips
Ghent University
Department of Telecommunications and Information Processing
Image Processing and Interpretation
Sint-Pietersnieuwestraat 41
9000 Ghent, Belgium
E-mail: vzlokoli@telin.ugent.be

Stefan Schulte
Etienne Kerre
Ghent University
Department of Applied Mathematics and Computer Science
Krijgslaan 281 (BLD 59)
9000 Ghent, Belgium

Abstract. *We propose a fuzzy logic recursive scheme for motion detection and spatiotemporal filtering that can deal with the Gaussian noise and unsteady illumination conditions in both the temporal and spatial directions. Our focus is on applications concerning tracking and denoising of image sequences. We process an input noisy sequence with fuzzy logic motion detection to determine the degree of motion confidence. The proposed motion detector combines the membership of the temporal intensity changes, appropriately using fuzzy rules, where the membership degree of motion for each pixel in a 2-D sliding window is determined by a proposed membership function. Both the fuzzy membership function and the fuzzy rules are defined in such a way that the performance of the motion detector is optimized in terms of its robustness to noise and unsteady lighting conditions. We simultaneously perform tracking and recursive adaptive temporal filtering, where the amount of filtering is inversely proportional to the confidence in the existence of motion. Finally, temporally filtered frames are further processed by a proposed spatial filter to obtain a denoised image sequence. Our main contribution is a robust novel fuzzy recursive scheme for motion detection and temporal filtering. We evaluate the proposed motion detection algorithm using two criteria: (1) robustness to noise and to changing illumination conditions and (2) motion blur in temporal recursive denoising. Additionally, we make comparisons in terms of noise reduction with other state of the art video denoising techniques.* © 2006 SPIE and IS&T. [DOI: 10.1117/1.2201548]

1 Introduction

In digital video, the general motion detection problem is quite complex because it is not always easy to distinguish illumination changes from real motion and because of the aperture problem (caused by a narrow field of view). As such, "close to perfect" solutions tend to be very time-consuming. However, in many applications it is sufficient to detect changes in the scene rather than actual motion, and even to detect only some of the changes. In this paper, we consider motion detection in terms of robust change detection against noise and slowly varying illumination conditions. This means that our method will not be able to distinguish completely, changes due to motion from other changes due to rapidly varying illumination, scene cuts, fast camera zoom, panning, etc. Despite these restrictions, there are numerous applications for this kind of motion detection scheme. It can be used for surveillance objectives, e.g., to monitor a room in which there is not supposed to be any motion, or the detection results can be a useful input for more advanced, higher level video processing techniques, such as the tracking of objects through time (e.g., Refs. 1–4). Other applications are noise removal[5–11] and deinterlacing[12,13] where one applies temporal filtering depending on the outcome of the motion detector.

Given the outlined restrictions, the main problem to be tackled is to distinguish image noise, caused by various sources, from real changes in picture intensity. Additionally, we try to adapt the method to changing illuminating conditions in both the spatial and the temporal directions. A trivial and very fast (but not very good) solution for pixel-by-pixel change detection is to simply subtract the gray levels of successive frames, and to conclude that the pixel has changed when the difference exceeds a given threshold. Because only one pixel is considered at a time, the computational cost is quite low.

While this technique works reasonably well at low noise levels, the performance degrades rapidly with increasing noise level. Also in practice, the threshold must be tuned to the noise level, which must be estimated on a global or

Paper 05138R received Jul. 20, 2005; revised manuscript received Oct. 25, 2005; accepted for publication Oct. 25, 2005; published online May 25, 2006. This paper is a revision of a paper presented at the SPIE conference on Image and Video Communications and Processing, Japan. 2005, San Jose, Calif. The paper presented the appears (unrefereed) in SPIE Proceedings Vol. 5685.
1017-9909/2006/15(2)/023008/13/$22.00 © 2006 SPIE and IS&T.

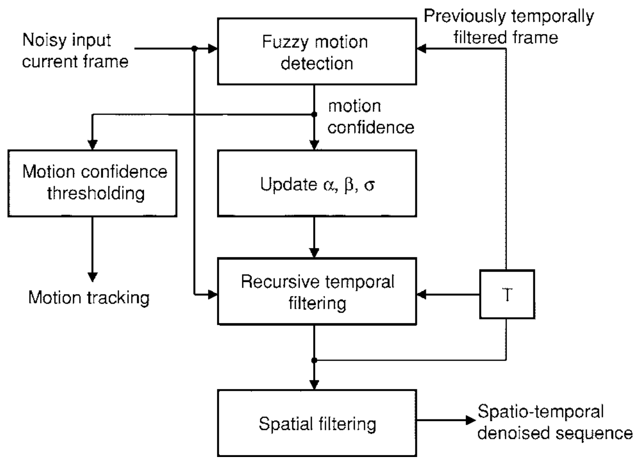

Fig. 1 Proposed algorithm framework.

Table 1 Common fuzzy intersection (conjunction) operators.

Fuzzy Intersection $A \cap B$ [Conjunction $\gamma_A(u) \wedge \gamma_B(u)$]	
Minimum	$\min[\gamma_A(u), \gamma_B(u)]$
Algebraic product	$\gamma_A(u)\gamma_B(u)$
Bounded product	$\max[0, \gamma_A(u)+\gamma_B(u)-1]$
Drastic product	$\begin{cases}\gamma_A(u), & \gamma_B(u)=1 \\ \gamma_B(u), & \gamma_A(u)=1 \\ 0, & \gamma_A(u), \gamma_B(u)<1\end{cases}$

local basis. To tackle the noise problem, some more sophisticated techniques are required. One could examine regions with more than one pixel, and define and calculate some characteristic functions (e.g., the spatial average in a window of pixels), as in Refs. 4, 14, and 15, at the expense of an increased complexity.

In this paper, we propose an alternative approach for motion detection and video denoising. We restrict ourselves to a recursive algorithm requiring a small number of computations.

The idea is to use adaptive thresholding, where the threshold is adapted to the local gray-scale statistics and to the spatial context of the pixel. The resulting method is highly insensitive to noise as well as to slow illumination changes. Moreover, it is locally adaptive, i.e., it adapts itself to spatially varying noise. In Ref. 16, some more complex techniques to obtain a threshold based on the noise or signal properties are proposed and applied to simple frame differencing. Our method is different, since it uses data coming from a longer period of time, and the threshold is adapted to both temporal and spatial information.

It is quite obvious that different applications require different approaches. In the case of motion detection for tracking, it is important that more or less only "real" motion is detected: the goal is to detect moving objects. This means, on the one hand, that if the input sequence is noisy, we do not want the noise to be labeled as motion, while on the other hand, it is not that important to detect every single changed pixel of an object. However, in the case of motion detection for denoising, where the detection result is used for temporal filtering, undetected changes can lead to (unwanted) motion blur, which is much more relevant to image quality than falsely labeled pixels due to noise. By using fuzzy logic we aim to define a confidence measure with respect to the existence of motion, called hereafter (the degree of) motion confidence. Based on the defined motion confidence, we optimize temporal denoising and motion detection.

We show the video denoising performance in terms of peak signal-to-noise ratio (PSNR); a newly defined objective measure; i.e., peak temporal signal-deviation-to-noise ratio (PTSDNR) and in terms of visual quality (subjective measure). Additionally we demonstrate the motion detection performance of the new technique in terms of the PTSDNR and its robustness to varying noise levels and illumination conditions.

Section 2 presents the general framework of the algorithm for simultaneous motion detection and video denoising, and we show the main blocks of the proposed algorithm. Subsequently, the proposed fuzzy logic recursive motion detector is explained in Sec. 3. In Sec. 4.1, we evaluate parameters used for the recursive temporal filtering and the spatiotemporal adaptive noise level detection, while in Sec. 4.2, a spatial filter is introduced. Finally, experimental results are given in Sec. 5 and conclusions are drawn in Sec. 6.

2 Novel Proposed Scheme For Simultaneous Motion Detection and Video Denoising

In this paper, a novel algorithm for simultaneous motion detection and video denoising is presented. The method is intended for processing noisy image sequences corrupted with white Gaussian noise, which is not necessarily stationary in space and time, and with spatiotemporal illumination changes. The proposed algorithm is depicted in Fig. 1.

We first process a noisy input sequence with the fuzzy logic motion detection using the current noisy (I_n) frame and the previous temporally processed (I_p) frame to determine the degree of motion confidence. The degree of motion confidence is expressed as a real number between two extremes: zero (no motion for sure) and one (motion for sure), in a fuzzy logic manner.[17,18] Specifically, the proposed fuzzy-logic-based motion detector combines several pixels from a 2-D neighborhood of the difference image:

$$\Delta(x,y,t) = |I_n(x,y,t) - I_p(x,y,t-1)|. \qquad (1)$$

The combination is performed in a fuzzy manner to determine the motion confidence $\theta(x,y,t)$ (explained in Sec. 3.2) for a certain spatiotemporal position (x,y,t).

Next, the estimated motion confidence is used for motion detection (tracking) and for video denoising. For detection of motion, we perform the following motion confidence thresholding:

Table 2 Common fuzzy union (disjunction) operators.

Fuzzy Union $A \cup B$ [Disjunction $\gamma_A(u) \vee \gamma_B(u)$]	
Maximum	$\max[\gamma_A(u), \gamma_B(u)]$
Algebraic sum	$\gamma_A(u)+\gamma_B(u)-\gamma_A(u)\gamma_B(u)$
Bounded sum	$\min[1, \gamma_A(u)+\gamma_B(u)]$
Drastic sum	$\begin{cases} \gamma_A(u), & \gamma_B(u)=0 \\ \gamma_B(u), & \gamma_A(u)=0 \\ 1, & \gamma_A(u), \gamma_B(u)>0 \end{cases}$

$$\mathrm{MD}(x,y,t) = \begin{cases} \text{true} & \theta(x,y,t) > \Omega \\ \text{false} & \text{otherwise}, \end{cases} \tag{2}$$

where the threshold Ω depends on the video sequence's resolution and on the application. By thresholding we determine two states of the motion detector: motion true and motion false. Subsequently, the estimated information can be used for motion tracking (in that case, motion information is combined with image segmentation) of the noisy video sequence. However, in this paper, we concentrate on denoising and reliable motion detection.

In the proposed algorithm, motion confidence $\theta(x,y,t)$ is used to determine the parameters $\alpha(x,y,t)$ and $\beta(x,y,t)$ for temporal recursive filtering and spatiotemporal adaptation of the algorithm to a changing noise level and illumination, respectively (see Sec. 4.1).

Based on the determined parameter $\beta(x,y,t)$ we perform a spatiotemporal adaptation of the noise variance $\sigma(x,y,t)$ for each pixel position (x,y,t). For the first frame only, we estimate the noise standard deviation using the method of Ref. 19 [for the whole frame, i.e., $\sigma(x,y,t)=\sigma_0$] and then adapt it throughout the sequence for each position $\sigma = \sigma(x,y,t)$, separately. This spatially local adaptation of noise standard deviation estimate $\sigma(x,y,t)$, is realized by recursive averaging. The higher is the degree of motion confidence $\theta(x,y,t)$, the closer is the current value $\sigma(x,y,t)$ to the previous value $\sigma(x,y,t-1)$:

$$\sigma(x,y,t) = [1-\beta(x,y,t)]\delta(x,y,t) + \beta(x,y,t)\sigma(x,y,t-1). \tag{3}$$

The parameter $\delta(x,y,t)$ represents a rough estimate of the local averaged changes (due to noise, motion, and illumination changes), at position (x,y,t), and is defined as follows:

$$\delta(x,y,t) = \frac{\sum_{i,j=-1}^{1} |\Delta(x+i,y+j,t)|}{9}. \tag{4}$$

The idea of Eq. (3) is to update $\sigma(x,y,t)$ at each time instant t, and for each spatial position (x,y), when no significant motion is detected (motion confidence is relatively small) and to prevent updating in case when motion confidence $[\theta(x,y,t)]$ is relatively large. This is done by changing $\beta(x,y,t)$.

Another parameter $\alpha(x,y,t)$ controls the amount of filtering based on the estimated noise level and the motion confidence. This parameter $\alpha(x,y,t)$ is directly proportional to the estimated motion confidence, and takes values $\alpha \in [0,1]$. As detailed in Sec. 4.1, we employ $\alpha(x,y,t)$ as a weighting factor within a classical recursive scheme:

$$I_p(x,y,t) = \alpha(x,y,t)I_n(x,y,t) + [1-\alpha(x,y,t)]I_p(x,y,t-1). \tag{5}$$

Note that the motion confidence $\theta(x,y,t)$ depends on the noise standard deviation from the previous frame $\sigma(x,y,t-1)$ (see Sec. 3.2), because the noise estimate $\sigma(x,y,t)$ is not known at that stage (see Fig. 1). Consequently, the parameters $\alpha(x,y,t)$ and $\beta(x,y,t)$ are also influenced to some degree by a $\sigma(x,y,t-1)$.

Subsequently, a spatial filter (Sec. 4.2) is applied to the temporally filtered frame (out of the recursive loop). Since the temporally filtered frames will contain nonstationary noise (in some areas of the image the remaining noise is stronger than in other areas), the proposed spatial filter aims at spatially local adaptation to the estimated (detected) noise level and image structures.

As we can see in Fig. 1, the temporal filter and the fuzzy logic motion detector work in a closed loop, i.e., in a recursive manner where at each step the result (performance) of each module is improved by the other. We develop a novel fuzzy logic recursive motion detection scheme (Sec. 3) that provides us with a reliable motion detection. Specifically, as the output of the motion detector we obtain motion confidence degree $\theta(x,y,t)$, which is used to determine the optimal parameters for the temporal filtering and adaptation of the algorithm to changing noise level and illumination changes.

3 New Fuzzy Logic Recursive Motion Detector

In our previous work[19] we introduced a binary motion detector, which can distinguish only between motion and no motion. This motion detector was used to control the spatiotemporal filter by switching the temporal filter on or off. A disadvantage of this hard switching was that in the case of higher noise levels, due to motion ambiguities, the filter introduced motion blurring and artifacts, resulting in a reduced denoising performance. This is due to the "hard" nature of the decision process (wrong decisions are propagated).

Recently, in Ref. 20, a recursive scheme for change detection applied to motion tracking in the presence of low noise levels was proposed, where only information from the current and from the previously processed (temporally accumulated) frame was used to detect motion, aiming at reducing memory requirements. However, the motion detector was binary, and consequently was not well suited for motion detection in the presence of higher noise levels and for video denoising.

In this paper, we propose a fuzzy logic recursive motion detector for both video denoising and tracking. The determined degree of the motion confidence is expressed as a

Table 3 Fuzzy rule set ($\wedge$, AND; $\vee$; OR; $\exists$, there exists; $\forall$, for all).

Rule	Antecedent		Consequent
R_1	$\exists^{K}_{i_1,j_1,i_2,j_2,i_3,j_3=-K}\,(BF_1 \wedge BF_2 \wedge BF_3 \wedge BF_4)$		$M(x,y,t)=$true
R_2	$\forall^{K}_{i_1,j_1,i_2,j_2,i_3,j_3=-K}\,(SF_1 \vee SF_2 \vee SF_3 \vee SF_4)$		$M(x,y,t)=$false
	$[(i_1,j_1)\neq(i_2,j_2)\neq(i_3,j_3)]\wedge[(i_{1,2,3},j_{1,2,3})\neq(0,0)]$		
Subfact	Fact	Subfact	Fact
SF_1	$\Delta(x,y,t)=$SMALL	BF_1	$\Delta(x,y,t)=$BIG
SF_2	$\Delta(x+i_1,\ y+j_1,\ t)=$SMALL	BF_2	$\Delta(x+i_1,\ y+j_1,\ t)=$BIG
SF_3	$\Delta(x+i_2,\ y+j_2,\ t)=$SMALL	BF_3	$\Delta(x+i_2,\ y+j_2,\ t)=$BIG
SF_4	$\Delta(x+i_3,\ y+j_3,\ t)=$SMALL	BF_4	$\Delta(x+i_3,\ y+j_3,\ t)=$BIG

real number between zero (no motion for sure) and one (motion for sure), instead of the binary logic decision, that is, motion or no motion. This provides more freedom (adaptation possibilities) concerning further processing tasks: recursive temporal filtering and spatiotemporal adaptation of the noise estimate, as explained in Sec. 4.1.

The main idea of the proposed motion detection algorithm is to sum the contributions concerning the motion presence for groups of temporal pixel differences from a small neighborhood. This is different from other more general motion detection techniques that are based on simply averaging all pixel differences or determining the number of pixel differences that are above a certain predefined threshold, in a small neighborhood. In the proposed technique, the contribution of the (temporal pixel difference) groups depends on both the local noise level estimate and the locally estimated reliability of the motion presence. The larger is the reliability relative to the estimated noise level, the larger will be the contribution from the corresponding group of pixel differences.

The proposed motion detection scheme is explained using a fuzzy logic framework, since fuzzy logic is suitable for describing processes with uncertainties (concerning motion in our case). The uncertainties at lower processing levels (temporal pixel differences) are combined in a specific way (using fuzzy rules) to obtain a meaningful result at a higher processing level (motion confidence). Namely, the main problem in motion detection is to distinguish between large pixel differences caused by noise and large pixel differences caused by motion. Actually, there will be a gradual transition from small differences caused by noise to very large differences caused by motion. Fuzzy set theory is a natural extension of the classical binary logic, where such kinds of gradual transitions are modeled by means of a proper membership function. Consequently, we argue that a fuzzy logic framework is more appropriate than a nonlinear filter scheme (which could also have been used) in this particular case. This is mainly because of the fuzzy logic solution flexibility and construction simplicity, concerning the combination problem of different uncertainties. Nevertheless, the fuzzy logic framework can be easily optimized to be of low implementation complexity and requiring a small number of calculations.

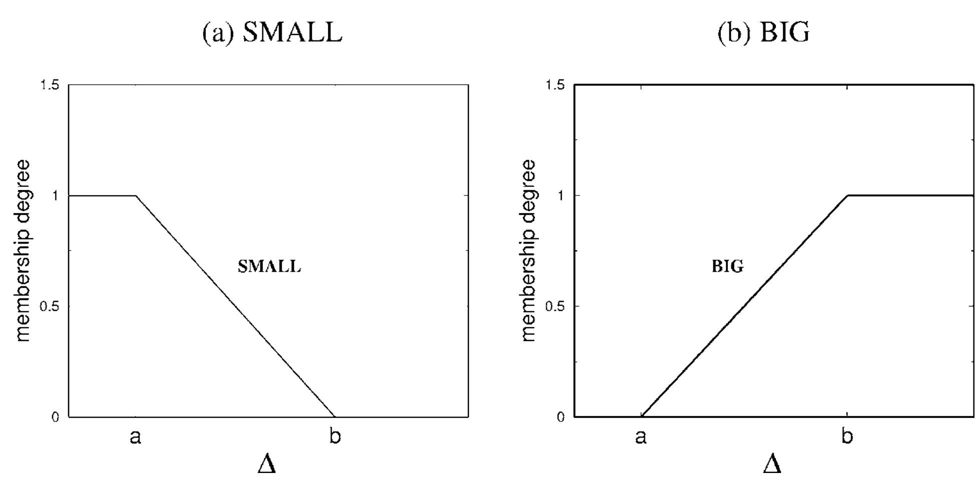

Fig. 2 Membership functions.

3.1 *Fuzzy-Logic-Based Information Processing*

Fuzzy set theory (fuzzy logic) is an extension of classical (crisp) set theory called binary logic.[17] It handles the concept of partial membership (partial truth). In fuzzy logic theory the membership degree can be taken as a value intermediate between zero and one. This is an extension of the black-or-white situation of classical set theory where no partial membership can be modeled. Fuzzy sets are generally represented by membership functions, i.e., a mapping from the underlying universe into the unit interval $[0,1]$.

The membership function of the intersection of two fuzzy sets A and B ($A \cap B$), in a universe U, is modeled by triangular norm[18,21] (see Table 1 for some examples), while the membership function of the union of two fuzzy sets A and B ($A \cup B$), in a universe U, is modeled by triangular co-norm[18,21] (see Table 2 for some examples). Additionally, we denote $\gamma_A(u)$ and $\gamma_B(u)$ as the corresponding membership in the fuzzy sets A and B, respectively, where u is an element of the universe U. For a more detailed description concerning fuzzy logic, refer to Refs. 22 and 23.

3.2 *Motion Detection*

In the proposed motion detection algorithm, particular groups of the temporal pixel differences (from a small spatial neighborhood) are analyzed and combined in a meaningful way to determine the motion confidence for each (central) pixel position (x,y,t). For each group of pixel differences $\Delta(x,y,t)$, we determine the local motion degree, which represents the contribution of the group, concerning the motion presence. Specifically, to determine the motion confidence for the central position (x,y,t), we first compute the membership degree for each temporal pixel difference in a spatial neighborhood (the membership correspond to the output of a membership function for the input temporal pixel difference). Subsequently, different combinations (defined by the fuzzy logic rules) of the membership degree for the central pixel position and the neighboring membership are evaluated, for each group of the temporal pixel differences. In the final step, such determined local motion for the groups are again combined by the defined fuzzy logic rule set (see Table 3), resulting in the motion confidence $\theta(x,y,t)$.

We define the membership functions SMALL and BIG for the difference signal $\Delta(x,y,t)$ [Eq. (1)] i.e., the membership $\gamma_S[\Delta(x,y,t)]$ and $\gamma_B[\Delta(x,y,t)]$, as follows:

$$\gamma_S[\Delta(;),a,b] = \begin{cases} 1 & \text{if } \Delta < a \\ 1 - \dfrac{\Delta - a}{b-a} & \text{if } a \leqslant \Delta \leqslant b \\ 0 & \text{if } \Delta > b, \end{cases} \tag{6}$$

$$\gamma_B[\Delta(;)a,b] = \begin{cases} 0 & \text{if } \Delta < a \\ \dfrac{\Delta - a}{b-a} & \text{if } a \leqslant \Delta \leqslant b \\ 1 & \text{if } \Delta > b, \end{cases} \tag{7}$$

which correspond to the plausibility of the statement "$\Delta(x,y,t)$ is SMALL" and "$\Delta(x,y,t)$ is BIG," respectively.

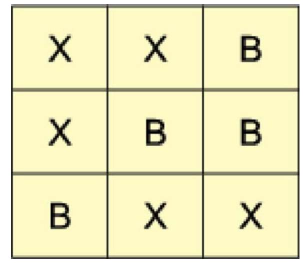

Fig. 3 Example of a 2D neighborhood sliding window for fuzzy motion detection: B, big; S, small; X, small or big.

Namely, γ_S increases for smaller $\Delta(x,y,t)$ values, while γ_B increases for higher $\Delta(x,y,t)$ values (see Fig. 2).

The parameters a and b determine the slope $\xi = 1/(b-a)$ of the membership functions, as shown in Fig. 2. Note that in this process, a and b are allowed to depend on x,y,t; i.e., we will evaluate $\gamma_S(x,y,t) = \gamma_S[\Delta(x,y,t), a(x,y,t), b(x,y,t)]$ and $\gamma_B(x,y,t) = \gamma_B[\Delta(x,y,t), a(x,y,t), b(x,y,t)]$.

The reason for using $a(x,y,t)$ and $b(x,y,t)$ is as follows. The higher the noise level, the larger the parameters a and b should be to avoid falsely detecting a significant amount of motion. Additionally, for a given noise level, we aim to make our motion detector more sensitive to motion foreground areas and less sensitive to areas that are less likely to undergo motion (motion background areas). To determine the probability of the presence of a motion area we use the locally averaged interframe difference $\delta(x,y,t)$ from Eq. (4). If $\delta(x,y,t)$ is relatively large in comparison to the noise level, we decide that the current pixel position belongs to the motion foreground area. In this case, we aim at increasing the sensitivity of the motion detector and thus we increase the slope ξ by reducing the value of parameter $b(x,y,t)$. Otherwise, we assume that the current pixel belongs to the motion background area and correspondingly reduce ξ by increasing $b(x,y,t)$ to reduce the sensitiveness of the motion detector. In such a way, we intend to reduce the amount of falsely detected motion pixels and simultaneously increase the amount of true positives (motion detected pixels).

To implement the explained idea we define $a(x,y,t)$ and $b(x,y,t)$, for each pixel position as follows:

$$a(x,y,t) = k_a \sigma(x,y,t-1), \tag{8}$$

and

$$b(x,y,t) = k_b \sigma(x,y,t-1) + \nu(x,y,t), \tag{9}$$

where $\nu(x,y,t)$ stands for a correction function defined as

$$\nu(x,y,t) = k_{b\delta} \frac{\sigma(x,\ y,\ t-1)}{1+\delta(x,y,t)} - k_{f\delta} \frac{\delta(x,y,t)}{1+\sigma(x,\ y,\ t-1)}, \tag{10}$$

where $k_a, k_b, k_{b\delta}$, and $k_{f\delta}$ are constants that are experimentally optimized in terms of optimal temporal compensation (minimum motion blur and highest PSNR) and in terms of the robustness of motion tracker against noise (minimum false motion detected pixels). Namely, k_a depends on the resolution and the frame rate of the input video (k_a

$\in [0,0.25]$) and k_b=4.2, where in the general case; $k_b \neq f(k_a)$. However, to avoid dependence of constant k_a on the particular type of sequence, we have fixed k_a=0.1 in our application for all sequences. Additionally, $k_{b\delta}$ and $k_{f\delta}$ denote adjustable constants for the motion background and motion foreground area, respectively. The default values of these parameters are $k_{b\delta}=2.5k_b=10.5$ and $k_{f\delta}=1.15k_b=4.83$.

As already mentioned in Sec. 2 at this stage we use the noise estimate $\sigma(x,y,t-1)$ from the previous time iteration [in Eqs. (8)–(10)], instead of the current noise level estimate (which is not available at the moment). This poses no problem, provided that the noise level changes from frame to frame relatively smoothly, which is most often the case in practice.

We not only consider the membership $\gamma_B(x,y,t)$ and $\gamma_S(x,y,t)$ at the current pixel position (x,y,t) but also at those of the specific spatial neighbors from the 2-D window $(K\times K)$, where K=3 in our implementation. We combine the values of $\gamma_B(x,y,t)$ and $\gamma_S(x,y,t)$ in a 2-D sliding window, by using the fuzzy rule set, shown in Table 3. Every rule determines whether there is motion $[M(x,y,t)=\text{true}]$ or not $[M(x,y,t)=\text{false}]$, in a fuzzy way.

The first rule in Table 3 states that there is motion $[M(x,y,t)=\text{true}]$ if the central signal difference $\Delta(x,y,t)$, in the 2-D window (e.g., Fig. 3), is BIG in a fuzzy way and at least three neighboring differences $\Delta(x+i,y+j,t)$ $(i,j=-1,0,1)$ are also considered as BIG. The first subfact BF_1 of the antecedent of the rule R_1, "$\Delta(x,y,t)$ is BIG," exist of a variable $\Delta(x,y,t)$ and a linguistic variable BIG. This fact is implemented by a membership function BIG, which produces membership degree $\gamma_B(x,y,t)$. Note that the membership degree is a number between 0 [the fact "$\Delta(x,y,t)$ is BIG" is not true for sure] and 1 [the fact "$\Delta(x,y,t)$ is BIG" is true for sure]. Hence, the activation degree of R_1 is just a conjunction of the four subfacts, which are combined by a chosen triangular norm (see Table 1). Specifically, the intersection $(\wedge)$ of all possible combinations of $\gamma_B(x,y,t)$ and three different neighboring BIG membership $\gamma_B(x+i,y+j,t)(i,j=-1,0,1)$ are computed, where the algebraic product (see Table 1) was used as the intersection operator. The results of these outcomes are summed, using the $\exists$ operator, which is the algebraic sum (see Table 2) of all instances (the local motion for the groups of the temporal pixel differences), to obtain the "true" motion confidence $M_{\text{true}}(x,y,t)$. In other words, the activation degree of the R_1 indicates to which degree $M(x,y,t)$ can be considered as true.

Similarly, the second rule states when there is no motion $[M(x,y,t)=\text{false}]$ as explained in Table 3, where $\forall$ stands for the algebraic product of all defined algebraic sums of the subfacts SF_1, SF_2, SF_3, and SF_4 (algebraic sum was used for the union operator $\vee$). The second rule in Table 3 represents the complement of the first rule and the output value of the second rule $M_{\text{false}}(x,y,t)$ corresponds to the "false" motion confidence.

Finally, we define the motion confidence for each pixel position (x,y,t) as follows:

Table 4 Pixels involved in the calculation of the fuzzy derivatives: each directions D (column 1) corresponds to a certain set of pixel positions (middle section) and last column specifies in which directions derivative pairs are found.

Direction, D	Pixel Positions			Derivative Pair
	N_1	C	N_2	
NW	$(x-1,\ y+1)$	(x,y)	$(x+1,\ y-1)$	$(-1,-1)$
N	$(x-1,\ y)$	(x,y)	$(x+1,\ y)$	$(0,-1)$
NE	$(x-1,\ y-1)$	(x,y)	$(x+1,\ y+1)$	$(1,1)$
W	$(x,\ y+1)$	(x,y)	$(x,\ y-1)$	$(-1,0)$
E	$(x,\ y+1)$	(x,y)	$(x,\ y-1)$	$(1,0)$
SW	$(x-1,\ y+1)$	(x,y)	$(x+1,\ y-1)$	$(1,1)$
S	$(x-1,\ y)$	(x,y)	$(x+1,\ y)$	$(0,1)$
SE	$(x-1,\ y+1)$	(x,y)	$(x+1,\ y-1)$	$(1,1)$

$$\theta(x,y,t)=\frac{M_{\text{true}}(x,y,t)}{M_{\text{true}}(x,y,t)+M_{\text{false}}(x,y,t)}. \tag{11}$$

Nevertheless, since the second rule is defined as a complement of the first rule, we can compute the motion confidence as $\theta(x,y,t)=M_{\text{true}}(x,y,t)$. Consequently, in such manner, we reduce the complexity and computation load of the method, since only the output from the first rule must be computed.

To summarize, we note that the determined motion confidence θ is dependent on both the local noise standard deviation σ and the temporal pixel difference Δ, i.e., $\theta(x,y,t)=f[\sigma(x,y,t_i),\Delta(x_i,y_i,t)]$, where (x,y,t) and (x_i,y_i,t_i) are coordinates of the central pixel position and the pixel positions from a spatial neighborhood, respectively. Further on, θ is also indirectly dependent on the parameters a, b, and δ, where $a(x,y,t)=f[\sigma(x,y,t)],b(x,y,t)=f[\sigma(x,y,t),\delta(x,y,t)]$, and $\delta(x,y,t)=f[\Delta(x_i,y_i,t)]$. Consequently, we introduce maximum two degrees of freedom for all parameters.

4 Parameter Evaluation and Spatial Filtering

In this section, we discuss the computation of the parameters $\alpha(x,y,t)$ and $\beta(x,y,t)$ for the recursive temporal filtering and the spatiotemporal adaptive noise level estimation, respectively Sec. 4.1. Additionally, we introduce the new fuzzy logic spatial filter Sec. 4.2, which is applied on the temporally filtered sequence, and which aims at filtering the remaining nonstationary noise.

4.1 *Parameter Optimization*

Using the degree of motion confidence $\theta(x,y,t)$, we determine the amount of temporal filtering, described by the parameter $\alpha(x,y,t)$. The idea of the recursive temporal filtering is as follows: when the degree of motion confidence is larger we filter less, while in the opposite case, we filter

Table 5 PSNR for processed sequences corrupted with the Gaussian noise σ_n=10.

Image Sequence	PSNR					
	FRSTF	MDAKNN	αTrimmed	Rational	MCWF	SEQWT
"Salesman"	33.9	32.5	29.4	30.6	32.9	34.1
"Deadline"	34.7	32.2	24.3	27.3	33.1	*X*
"Miss America"	36.5	35.3	35.1	35.2	35.9	37.5
"Trevor"	34.3	34.1	33.5	34.3	35.3	*X*
"Tennis"	31.2	30.5	24.5	26.5	31.3	32.2

more. Namely, we control the amount of filtering through the parameter $\alpha(x,y,t)$, which is directly proportional to the estimated motion confidence and takes values of $[0,1]$. Note that for one extreme case, $\alpha(x,y,t)=1$, there is no averaging at all, while for the other extreme case, $\alpha(x,y,t)=0$, only the processed (filtered) pixel from the previous temporal position $t-1$ is substituted. The smaller the α, the stronger the filtering will be performed but also the more information from the previous frame will be taken into account. Thus, in case when $\alpha(x, y, t-1)$ is relatively large $[\alpha(x, y, t-1)\approx 1]$ and $\theta(x,y,t)$ relatively small $[\theta(x,y,t)\approx 0]$, the amount of filtering will be determined as $\alpha(x,y,t)\approx 0$. Hence, the noisy pixel from the previous frame would be substituted, i.e., the noise will propagate through the processed sequence.

To refine our method to be robust against noise propagation, we determine $\alpha(x,y,t)$ not only according to the motion confidence in the current frame $[\dot{\theta}(x,y,t)]$, but also according to the amount of the temporal filtering in the previous frame, $\alpha(x, y, t-1)$. For that, we found one particular solution, defined as follows:

$$\alpha(x,y,t) = \frac{1}{2}\alpha^2(x, y, t-1) + \left[1 - \frac{1}{2}\alpha(x, y, t-1)\right]\alpha'(x,y,t), \qquad (12)$$

where $\alpha'(x,y,t)$ is an initial estimate based on the motion confidence $\theta(x,y,t)$ only, that is, $\alpha'(x,y,t) = 1.15[\theta(x,y,t)]^{1/2}$ [for $\alpha'(x,y,t)\geqslant 1$ we assign $\alpha'(x,y,t) = 1$]. As a result, we obtain a smoother transition of the amount of temporal filtering from one frame to the other. Note that by employing Eq. (12), we both prevent the noise propagation in time (as already explained) and enable strong filtering provided by the recursive schemes [e.g., in case when in two consecutive time instances no motion has been detected $\alpha'(x,y,t)\approx 0$ and $\alpha(x, y, t-1)\approx 0$] as well. Additionally, in the case when in the previous frame no filtering was performed $[\alpha(x, y, t-1)\approx 1]$ and when no motion was detected in the current frame $[\theta(x,y,t)=0]$, the determined amount of the filtering will be $\alpha(x,y,t)\approx 0.5$, i.e., simple averaging is done.

As defined in Eq. (3) the updated value of $\sigma(x,y,t)$ depends on $\delta(x,y,t)$ [defined in Eq. (4)] and the weighting parameter $\beta(x,y,t)$, which controls the updating (averaging) of $\sigma(x,y,t)$. We found experimentally that the following correlation between $\beta(x,y,t)$ and the motion confidence $\theta(x,y,t)$:

$$\beta(x,y,t) = 1.5[\theta(x,y,t)]^{1/2}, \qquad (13)$$

provides reliable results, where $\beta(x,y,t)=1$ in case of $\beta(x,y,t)>1$. As we can observe the parameters $\alpha'(x,y,t)$ and $\beta(x,y,t)$ are defined so as to have the same dependence on θ with the only difference being the constant, i.e., $\alpha'(x,y,t)=1.31\beta(x,y,t)$. With this we intended to reduce the number of of freedom for the parameters used in the proposed method. Further on, in our experiments, we discovered that more advantageous performance is obtained in terms of adaptation to varying noise levels by locally averaging the estimated $\sigma(x,y,t)$ value within a 2-D sliding window (contains half of the values from the previous and half from the current frame) centered at (x,y,t).

4.2 *Spatial Filter*

The general idea behind the proposed spatial filter is to filter the nonstationary noise left by the preceding temporal filter. This is done by a local spatial filtering, which is adaptive to image structures and noise level present in the corresponding spatial neighborhood. Hence, the aim of the proposed filter is to distinguish between the local variations due to noise and the local variations that are due to image structures. The proposed spatial filter performs weighted threshold averaging of the temporally processed frame, where the weighting coefficients are based on fuzzy derivatives.[24] However, instead of defining fuzzy derivatives in terms of negative, positive, big, and small, as described in Ref. 24, we evaluate the corresponding directional derivatives based only on their absolute value, i.e., in terms of small and big.

Consider a 3×3 neighborhood of a pixel $I_p(x,y,t)$. The derivative $\nabla_D(x,y,t)$ is defined as the absolute difference between the central pixel (x,y,t) and its neighbor in the direction D; $D\in\{\mathrm{NW,W,SW,S,SE,E,NE,N}\}$ [e.g., $\nabla_N(x,y,t)=|I_p(x, y-1, t)-I_p(x,y,t)|$].

Table 6 PSNR for processed sequences corrupted with the Gaussian noise σ_n=15.

Image Sequence	PSNR					
	FRSTF	MDAKNN	αTrimmed	Rational	MCWF	SEQWT
"Salesman"	31.9	29.2	28.4	29.5	30.8	31.1
"Deadline"	32.2	28.7	23.7	32.9	30.6	*X*
"Miss America"	34.6	30.2	32.7	26.6	33.6	36.0
"Trevor"	32.3	29.7	31.5	32.1	32.8	*X*
"Tennis"	29.2	28.1	23.5	26.2	29.3	30.1

We assume that in case an edgelike image structure extends in a certain direction D, it causes large derivative values perpendicular to the direction D at the current pixel position (x,y) and at the neighboring pixels as well [e.g., for edge-structure in W to E direction, $\nabla_N(x,y,t)$, $\nabla_N(x,\ y-1,\ t)$, and $\nabla_N(x,\ y+1,\ t)$ are expected to be relatively big]. In such a manner, by taking into account those three derivative values for each direction (and combining them in a fuzzy logic manner), we distinguish between local variations due to noise and due to the edgelike image structures. In Table 4 we give an overview of the pixels that are involved in the calculation of the derivatives.

We use the following logic to determine the fuzzy derivative value for a certain direction D and (x,y,t) coordinate position. If the central derivative $\nabla_D^C(x,y,t)$ is small, AND either of the two neighboring derivatives, $\nabla_D^{N_1}(x,y,t)$ for neighbor N_1 OR $\nabla_D^{N_2}(x,y,t)$ for neighbor N_2 is SMALL, we conclude that there is small derivative value in direction D, i.e., the small edge magnitude along the direction perpendicular to D. Specifically, we define the D'th-direction fuzzy derivative value with fuzzy rule 1.

4.2.1 *Fuzzy rule 1*

IF {[$\nabla_D^C(x,y,t)$ is SMALL AND $\nabla_D^{N1}(x,y,t)$ is SMALL]

OR [$\nabla_D^C(x,y,t)$ is SMALL AND $\nabla_D^{N2}(x,y,t)$ is SMALL]}

THEN $\nabla_D(x,y,t)$ is SMALL,

where the union (AND) and intersection (OR) operators are defined using algebraic product and algebraic sum (see Tables 1 and 2), respectively. The fuzzy membership function SMALL, used in fuzzy rule 1, decreases for larger ∇_D, and is defined as follows:

$$F_S[\nabla_D(;),T_s] = \begin{cases} 1-\dfrac{\nabla D}{Ts} & \text{if } \nabla_D < T_s \\ 0 & \text{otherwise} \end{cases} . \qquad (14)$$

The parameter T_s is adapted to the local noise level [$T_s=2.5\sigma(x,y,t)$], and the actual value of the statement "$\nabla_D(x,y,t)$ is SMALL" is $F_S[\nabla_D(x,y,t),T_s]$. The choice of $T_s=2.5\sigma(x,y,t)$ was experimentally found to be best in terms of mean square error for the denoised sequence in correspondence to the original one.

We define the proposed weighted averaging (spatial) filter in terms of the determined derivatives (fuzzy rule 1), for each of the eight neighbors, as follows:

$$I_{\text{out}}(x,y,t) = \frac{\sum_{k=-1}^{1}\sum_{l=-1}^{1} I_p(x_k,y_l,t)w(x_k,y_l,t)}{\sum_{k=-1}^{1}\sum_{l=-1}^{1} w(x_k,y_l,t)}, \qquad (15)$$

where $x_k=x+k$ and $y_l=y+l$, and with the weighting coefficients w defined as the output of the fuzzy controller defined by fuzzy rule 1 for each neighbor, i.e., the weight for the D'th direction is equal to $F_S[\nabla_D(x,y,t),T_s]$. The smaller the derivative $\nabla_D(x,y,t)$ defined by the fuzzy rule 1, the larger will be the weighting coefficients [see Eq. (14)] and hence the corresponding neighboring pixel values will be taken more into account for the spatial filtering.

5 Experimental Results

In this section, we demonstrate the performance of the proposed algorithm for video denoising and motion tracking. The tested (original) video sequences were distorted by various levels of input Gaussian noise ($\sigma_n=5,10,15,20,25$). Additionally, video sequences with unsteady lighting conditions and noise levels varying in time throughout the sequence were considered. The camera used for obtaining the sequences was fixed. This is typical in applications such as video surveillance, where motion tracking and video quality improvement are performed.

We first tested several video sequences corrupted with steady Gaussian noise throughout the input corrupted sequences and with steady lighting conditions, such as, "Salesman," "Trevor," "Miss America," "Bicycle," "Chair," "Tennis," "Deadline," "Flower Garden," and "Miss America." We corrupted them with various noise levels—$\sigma_n=5,10,15,20,25$—and compared the denoising results with several state of the art techniques.

The proposed fuzzy logic recursive spatiotemporal filtering technique (FRSTF) was compared with several state-of-the-art techniques for video denoising: The rational filter[25] (rational), the motion-detail adaptive K-NN filter[19] (MDAKNN), the spatiotemporal sequential filter of Ref. 10

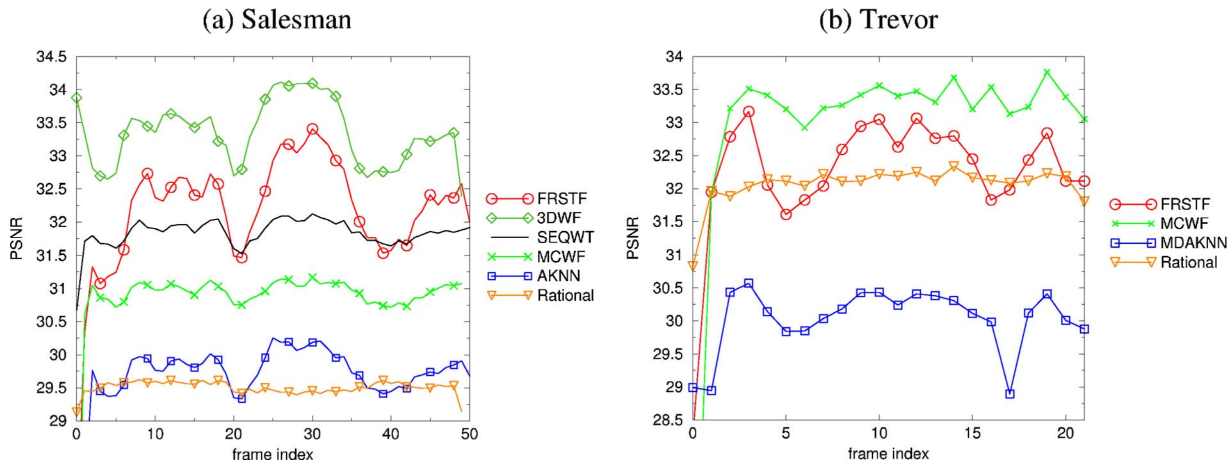

Fig. 4 PSNR per frame for the input noise level $\sigma_n=15$.

(SEQWT), the multiclass wavelet based spatio-temporal filter[11] (MCWF), the 3-D wavelet filter[26] (3DWF), and the α-trimmed filter[27] (α trimmed).

In our experiments, we used the following two objective quality measures for the evaluation of the noise reduction performance for the proposed temporal and the spatio-temporal filter (per frame t): (1) PSNR, determined as follows:

$$\mathrm{PSNR}(t)(\mathrm{dB}) = 20\log_{10}\frac{S_p}{\left[\frac{1}{MN}\sum_x\sum_y e(x,y,t)\right]^{1/2}}, \tag{16}$$

where $e(x,y,t)=[I_{\mathrm{out}}(x,y,t)-I_{\mathrm{orig}}(x,y,t)]^2$, with I_{out} and I_{orig} being the corresponding pixel values of the spatiotemporally filtered and the original sequence; and (2) PTSDNR, which is defined in the following way:

$$\mathrm{PTSDNR}(t)(\mathrm{dB}) = 20\log_{10}\frac{S_p}{\left[\frac{1}{MN}\sum_x\sum_y e^T(x,y,t)\right]^{1/2}}, \tag{17}$$

where $e^T(x,y,t)=(\Delta^t_{\mathrm{orig}}-\Delta^t_{\mathrm{out}})^2$, with $\Delta^t_{\mathrm{orig}}=|I_{\mathrm{orig}}(x,y,t)-I_{\mathrm{orig}}(x,y,t-1)|$ and $\Delta^t_{\mathrm{out}}=|I_{\mathrm{out}}(x,y,t)-I_{\mathrm{out}}(x,y,t-1)|$. The peak signal value in our case is $S_p=255$, since we have 8 bits to store a single gray-scale value per each pixel position in the gray-scale image sequences, hence having pixel values in the range $[0,255]$. The parameters M and N stand for the width and height of the image sequence.

The first measure, PSNR [Eq. (16)] indicates how well the noise is smoothed and the structures preserved on average in the spatial domain, in terms of the average correlation of the filtered and the original image sequence, at each time instant. On the other hand, the novel PTSDNR [Eq. (17)] measure describes the correlation of the temporal gradients of the filtered and original image sequence, providing us with the information about noise variation/

Table 7 PSNR for the temporally filtered sequences corrupted with the Gaussian noise $\sigma_n=10, \sigma_n=15$, and $\sigma_n=20$.

Image Sequence	PSNR					
	$\sigma_n=10$		$\sigma_n=15$		$\sigma_n=20$	
	B	*F*	*B*	*F*	*B*	*F*
"Salesman"	31.4	32.4	28.3	29.4	25.4	27.2
"Deadline"	31.7	32.9	28.5	29.6	26.2	27.5
"Miss America"	32.1	32.9	28.9	29.8	26.6	27.7
"Trevor"	30.7	31.3	27.6	28.5	25.4	26.1
"Tennis"	29.8	30.8	26.9	28.2	25.4	26.3

Table 8 PTSDNR for the temporally filtered sequences corrupted with the Gaussian noise $\sigma_n=10, \sigma_n=15$, and $\sigma_n=20$.

Image Sequence	PTSDNR					
	$\sigma_n=10$		$\sigma_n=15$		$\sigma_n=20$	
	B	*F*	*B*	*F*	*B*	*F*
"Salesman"	32.5	33.2	29.8	31.1	27.7	29.4
"Deadline"	32.5	34.6	29.4	31.6	27.2	29.8
"Miss America"	33.8	35.1	30.6	32.2	28.2	30.5
"Trevor"	30.8	31.6	28.2	29.6	26.1	27.2
"Tennis"	29.5	30.7	27.1	28.7	25.4	27.1

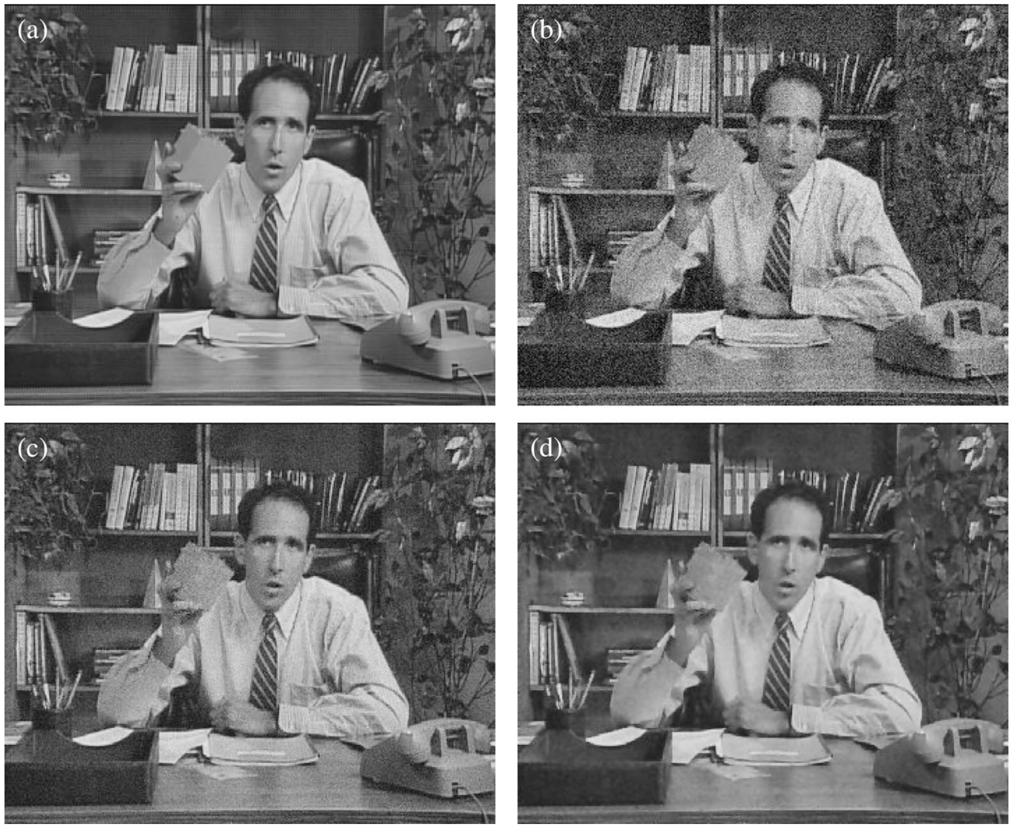

Fig. 5 Results for the 30th frame of "Salesman" sequence corrupted with a noise level σ_n=15: (a) original, (b) noisy, (c) temporal filter, (d) and spatiotemporal filter.

compensation in the temporal domain (which is usually very annoying for the human eye and yet not taken into account by the well known PSNR measure).

In Tables 5 and 6 we compare the proposed (spatiotemporal) filter performance in terms of the averaged PSNR over all filtered frames in the sequence for the noisy sequences with artificial uniformly spread Gaussian noise, with $\sigma_n=10$ and $\sigma_n=15$, respectively. We see that in comparison to the single resolution techniques (the rational filter, the MDAKNN filter, and the α-trimmed filter[27]), the proposed FRSTF method performs always significantly better (the average improvement is approximately 1.5 dB), while it is comparable to multiresolution methods: the SEQWT filter and the MCWF. Additionally in Fig. 4, we show the PSNR per frame for the "Salesman" and the "Trevor" sequence with additive Gaussian noise (σ_n=15), respectively, where the advantageous performance of the proposed method (FRSTF) can also be seen.

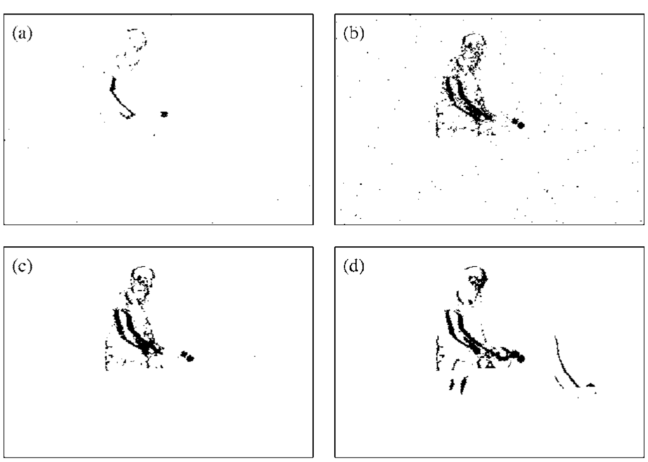

Fig. 6 Motion detection comparison (σ_n=20): (a) (σ_n=20) NRCD, (b) (σ_n=20) BMD [Tb=$2\sigma(x,y,t)$], (c) (σ_n=20) FMRD (Ω=0,75), and (d) (noise-free) FMRD (Ω=0.75).

To make a comparison between the proposed fuzzy solution and the corresponding binary one, in terms of motion detection and noise reduction, we define the binary version of the proposed fuzzy logic motion detection, i.e., we define the binary motion confidence $\theta_b(x,y,t)$, as follows:

$$\theta_b(x,y,t)=\begin{cases} 1 & \text{if } \Delta(x,y,t)>T_b(x,y,t) \text{ AND} \\ & \Delta(x+i,\ y+j,t)>T_b(x,y,t) \\ & \text{for at least three neighbors} \\ & i,j\in\{-1,0,1\} \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where the threshold value $T_b(x,y,t)=[a(x,y,t)+b(x,y,t)]/2$ is determined in accordance with the fuzzy functions defined in Eqs. (6) and (7). Namely, we say that for the (x,y,t) pixel position, the motion in this (binary) case is present (MD = true) when $\theta_b(x,y,t)=1$ and absent (MD = false) when $\theta_b(x,y,t)=0$.

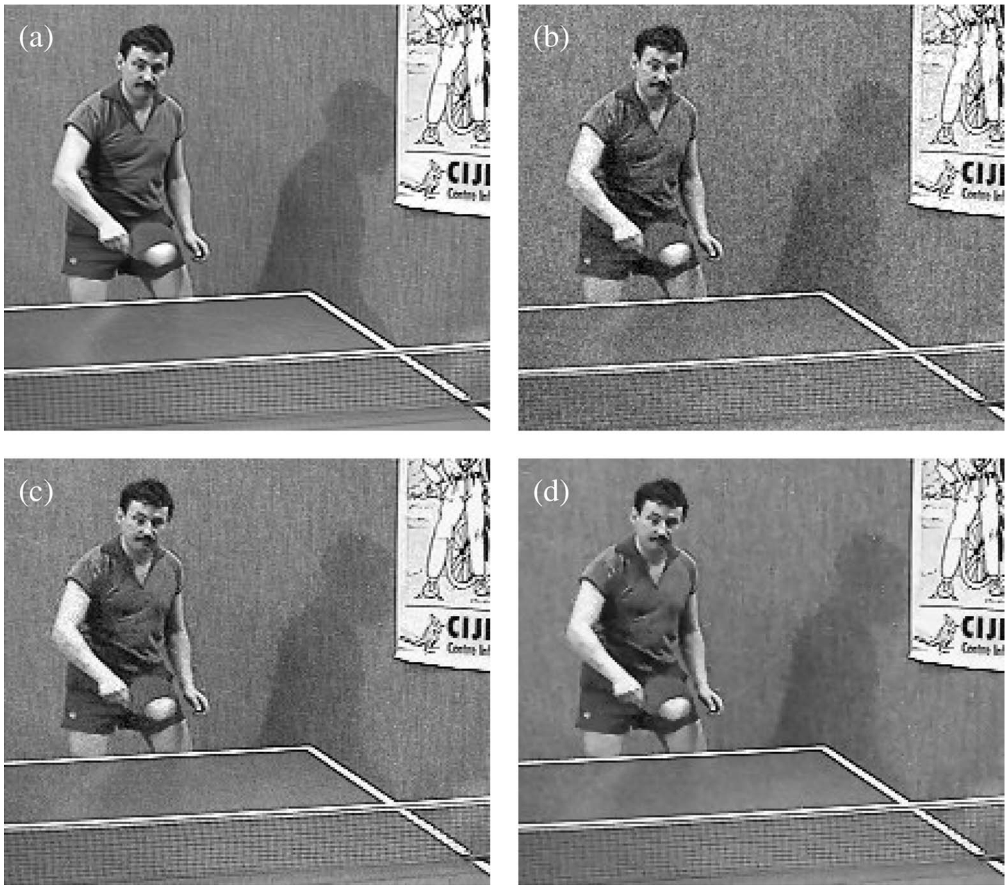

Fig. 7 Results for the 124th frame of "Tennis" sequence corrupted with a noise level σ_n=15: (a) original, (b) noisy, (c) temporal filter, (d) spatiotemporal filter.

We demonstrate the performance of the proposed temporal filter that uses fuzzy logic motion detection and corresponding fuzzy logic temporal filtering, in comparison to the binary motion detection version, as defined in Eq. (18), where the corresponding temporal filtering is based on averaging or no averaging at all. To show the comparison, we evaluate the PSNR and PTSDNR of the temporally filtered noisy image sequence with fuzzy (F) and binary (B) logic. The corresponding experimental results, in terms of PSNR and PTSDNR, averaged in time (through all sequences frames) are shown in Tables 7 and 8, respectively.

We conclude from the Table 7 that in terms of PSNR of the temporally filtered image sequences the fuzzy (F) logic approach provides on average 0.9 dB better results than the binary (B) approach, where the improvement is approximately the same for all noise levels. From Table 8 we conclude that, in terms of PTSDNR, the fuzzy (F) logic ap-

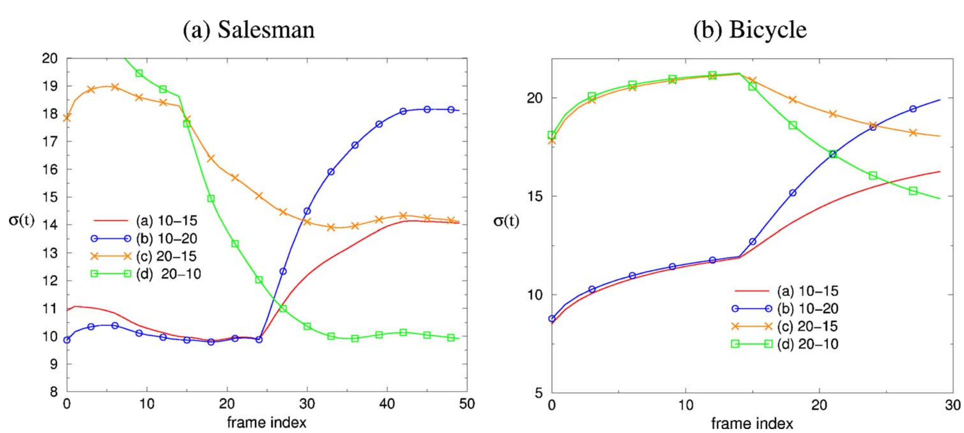

Fig. 8 Adaption of $\sigma(t)$.

proach provides on average 1.25 dB better results than the binary (B) approach, where the improvement is smaller for the lower noise levels ($\sigma_n=10$) 0.7 dB and higher for the higher noise levels ($\sigma_n=20$) 2 dB.

The temporal only and the spatiotemporal processed sequences by the proposed fuzzy logic algorithm, with added Gaussian noise of $\sigma_n=5,10,15,20,25$, can be viewed on http://telin.ugent.be/~vzlokoli/JEI/UNSP/. The video sequences illustrating the detected motion (tracking) can also be viewed there. In these sequences (used only for illustrative purposes) the detected motion was determined by the motion confidence thresholding (block) in Fig. 1 and described by Eq. (2), with threshold values $\Omega=0.5$ and $\Omega=0.75$. The motion detection sequences illustrate the sensitivity of the motion detector to noise (false detected motion) and temporal sequence changes (true detected motion), while the filtered (temporal and spatio-temporal) sequences illustrate robustness to motion blur and ability of efficient noise removal.

In Figs. 5 and 6 we show the denoising results for one (124th) frame of the "Tennis" sequence and one (30th) frame of the "Salesman" sequence, corrupted with $\sigma_n=15$, respectively. The figures illustrate the performance of the two steps of the proposed video denoising algorithm, that are the resulting temporally only filtered frame (left-down in Figs. 5 and 6) and the spatiotemporally filtered frame (right-down in Figs. 5 and 6). Additionally, the up-left and up-right image in Figs. 5 and 6 stand for the original and the noisy frame of the image sequence.

In Fig. 7 we compare the proposed motion detector for the "Tennis" sequence ($t=131$),with added noise of $\sigma_n=20$, with the noise robust change detection (NRCD) of Ref. 20 and with the result of the proposed method with the binary (BMD) logic. The fuzzy (FMRD) logic method performed on the same sequence without noise added was used in our experiment as a substitution for the groundtruth concerning motion. From Fig. 7, we can observe that the proposed FMRD method is always better, that is, more robust against noise (less false positive motion detected pixels) in comparison to the binary (logic approach) motion detection (BMD). Moreover, the FMRD method detects more motion (more true positive motion detected pixels) in comparison to the NRCD method, by considering the ground truth motion information from Fig. 7(d). By comparing Figs. 7(b) and 7(c), we conclude that the fuzzy logic motion detection (FMRD) performs better than the binary logic motion detection (BMD). The fuzzy logic approach detects the same or more "true" motion pixels while producing much less "false" motion detection pixels.

Nevertheless, by changing the threshold value Ω we can adapt the algorithm to the desired amount of detection, whereas in the other methods[16,20] the output is fixed. As the noise increases, number of true detected pixels decreases very slowly for the cost of efficient reduction of false (noisy) motion detected pixels. However, even for a relatively small threshold (Ω) and relatively high noise levels ($\sigma_n \approx 20$), the proposed algorithm provides reliable motion detection results.

We also tested our video denoising algorithm on video sequences with additive Gaussian noise that changes throughout the sequence in the temporal domain. As an experiment, we added noise, $\sigma_n=10$, in the "Salesman" and the "Bicycle" sequences, for the first 25 and 15 frames, respectively, and then changed the noise level to (a) $\sigma_n=15$ and (b) $\sigma_n=20$. Similarly, in another experiment $\sigma_n=20$ for the first 25 and 15 frames, for the "Salesman" and the "Bicycle" sequences, respectively, and then σ_n is changed to (c) $\sigma_n=15$ and (d) $\sigma_n=10$. Figure 8 shows how the estimated average $\sigma(t)$ value (averaged over 1 frame) changes from frame to frame and how large the slope is, i.e., the speed of the temporal adaptation. Specifically, in Fig. 8(a) the σ adaptation is illustrated for the "Salesman" sequence and cases (a) to (d), while in Fig. 8(b), the σ adaptation is illustrated for the "Bicycle" sequence and cases (a) to (d). The corresponding processed sequences can be seen on the Web at http://telin.ugent.be/~vzlokoli/JEI/NUNSP/CN, where both the denoised sequences and the motion detection results are shown. The value for the motion confidence thresholding (Fig. 1) [Eq. (2)] was $\Omega=0.75$. We can observe from the sequences that the motion detection adapts to the noise level change gradually and corrects its performance in a few frames (usually 1 to 4).

Finally, we tested our algorithm for motion detection and video denoising on video sequences with varying illumination changes in both the spatial and the temporal domains. For that, we experimented with a fixed camera and changed the illumination conditions in time and space. Namely, we performed an experiment in a dark room where the different lighting conditions were made by varying spatiotemporal changes of an additional light source. Additionally we added Gaussian noise $\sigma_n=15$ to the sequences and processed them. The tested sequences consist of a person coming into the room, moving, and after that leaving the room. The sequences illustrating the detected motion and the corresponding denoised sequences can be seen on the website at http://telin.ugent.be/~vzlokoli/JEI/NUNSP/CI/. The results demonstrate good robustness against slow illumination changes, while for the sudden changes, we conclude that more time is required for the algorithm to adapt.

6 Conclusion

A new adaptive recursive scheme for the fuzzy-logic-based motion detection and video denoising was proposed. The proposed fuzzy logic motion detection is robust against noise and slowly varying illumination changes. The reliable motion detection scheme enabled efficient temporal recursive filtering, which in turn improved the motion detection performance in the proposed recursive scheme. In the future, we aim to investigate the usage of color and spatial image features for a more efficient motion detection that could cope with fast illumination changes and fast zooming and panning.

Acknowledgments

A. Pižurica is postdoctoral researcher of FWO (Fonds voor Wetenschappelijk Onderzoek) Flanders.

References

1. J. Denzler and D. W. R. Paulus, "Active motion detection and object tracking," in *Proc. ICIP*, pp. 635–639 (1994).
2. C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. CVPR*, pp. 246-252 Fort Colins, CO (1999).

3. P. Kaew, Tra Kull Pong, and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” in *Proc. 2nd Workshop on Advanced Based Surveillance Systems* (2001).
4. M. Xu and T. Elis, “Illumination-invariant motion detection using colour mixture models,” in *Proc. British Machine Vision Conf.*, Manchester, UK (2001).
5. J. Drewery, R. Storey, and N. Tanton, “Video noise reduction,” in BBC Research Departement Report BBC RD 1984/7 (1984).
6. J. C. Brailean, R. Kleihorst, S. Efstratidis, K. Katsaggeleos, and R. Lagendijk, “Noise reduction filters for dynamic image sequences: review,” *Proc. IEEE* **83**, 1272–1292 (Sept. 1995).
7. K. Jostschulte, A. Amer, M. Schu, and H. Schroder, “Perception adaptive temporal TV-noise reduction using contour preserving prefilter techniques,” *IEEE Trans. Consum. Electron.* **44**(3), 1091–1096 (1998).
8. O. Ojo and T. Kwaaitaal-Spassova, “An algorithm for integrated noise reduction and sharpness enhancement,” *IEEE Trans. Consum. Electron.* **46**(3), 474–480 (2000).
9. G. De Haan, Video Processing for Multimedia Systems, University Press Eindhoven, The Nederlands (2003).
10. A. Pižurica, V. Zlokolica, and W. Philips, “Noise reduction in video sequences using wavelet-domain and temporal filtering,” *Proc. SPIE* **5266**, 48–59 (2003).
11. V. Zlokolica, A. Pižurica, and W. Philips, “Video denoising using multiple class averaging with multiresolution,” in *Proc. Int. Workshop VLBV03*, Lecture Notes in Computer Science, Springer Verlag, Madrid (2003).
12. D. V. D. Ville, Phd Dissertation, Ghent University (2001).
13. E. Bellers and G. De Haan, De-Interlacing: A key Technology for Scan Rate Conversion, Elsevier Science, Amsterdam (2000).
14. R. Jain and K. Skifstad, “Illumination independent change detection from real world image sequences,” *Comput. Vis. Graph. Image Process.* **46**, 387–399 (1989).
15. Y. Z. Hsu, H. H. Nagel, and G. Refers, “New likelihood test methods for change detection in image sequences” *Comput. Vis. Graph. Image Process.* **26**, 73–106 (1984).
16. P. L. Rosin, “Thresholding for change detection,” in *Proc. ICCV*, pp. 274–279, Bombay (1998).
17. E. Kerre, *Introduction to the basic principles of fuzzy set theory and some of its applications*, Communications & Cognition, Gent, Belgium (1991).
18. E. Kerre, *Fuzzy Sets and Approximate Reasoning*, Xian Jiaotong University Press (1998).
19. V. Zlokolica and W. Philips, “Motion and detail adaptive denoising of video,” *Proc. SPIE* **5298**, 403–412 (2004).
20. M. De Geyter and W. Philips, “A noise robust method for change detection,” in *Proc. ICIP*, Barcelona (2003).
21. E. Klement, R. Maisar, and E. Pap, *Triangular Norms*, Kluwer Academic Pub. (2000).
22. E. Cox, “Fuzzy fundamentals,” *IEEE Spectrum* **29**, 58-61 (Oct. 1992).
23. C. Lee, “Fuzzy logic in control systems: fuzzy logic controller,” *IEEE Trans. Syst. Man Cybern.* **20**, 319–323 (Mar. 1990).
24. D. Van De Ville, M. Nachtegael, D. Van der Weken, and E. Kerre, “Noise reduction by fuzzy image filtering,” *IEEE Trans. Fuzzy Syst.* **11**, 429–436 (Aug. 2003).
25. F. Cocchia, S. Carrato, and G. Ramponi, “Design and real-time implementation of a 3-D rational filter for edge preserving smoothing,” *IEEE Trans. Consum. Electron.* **43**, 1291–1300 (Nov. 1997).
26. W. Selesnick and K. Li, “Video denoising using 2D and 3D dual-tree complex wavelet transforms,” *Proc. SPIE* **5207**, 607–618 (2003).
27. J. Bednar and T. L. Wat, “Alpha-trimmed means and their relationship to median filters,” *IEEE Trans. Acoust., Speech, Signal Process.* **32**, 145–153 (Feb. 1984).

Vladimir Zlokolica received his MSc degree in electrical engineering in 2001 from the University of Novi Sad, Serbia and Montenegro. Since September 2001, he has been with the Department of Telecommunications and Information Processing, Ghent University, where he is a member of the group for image processing and interpretation, and is working toward his PhD degree with a thesis on nonlinear techniques for video denoising under the supervision of promotor Prof. W. Philips and copromotor Aleksandra Pizurica. His research interests include video processing, motion estimation, multiresolution representations, noise estimation, and objective quality assessment of video.

Aleksandra Pižurica received her diploma degree in electrical engineering from the University of Novi Sad in 1994, her MSc degree in telecommunications from the University of Belgrade, Serbia and Montenegro in 1997, and her PhD degree in applied sciences from the Ghent University, Belgium, in 2002. She is currently a postdoctoral researcher with the Department of Telecommunications and Information Processing of the Ghent University. Her research interests include statistical signal and image modeling, multiresolution representations, signal detection and estimation, and video processing.

Wilfried Philips received his diploma degree in electrical engineering in 1989 and his PhD degree in applied sciences in 1993, both from Ghent University, Belgium. From October 1989 to October 1997 he was with the Department of Electronics and Information Systems of Ghent University for the Flemish Fund for Scientific Research (FWO-Vlaanderen), first as a research assistant and later as a postdoctoral research fellow. Since November 1997 he has been with the Department of Telecommunications and Information Processing of Ghent University, where he is currently a full-time professor. His main research interests are image and video restoration, image analysis, lossless and lossy data compression of images and video, and processing of multimedia data.

Stefan Schulte received his MSc degree in computer science from the University of Ghent in 2003. Since September 2003, he has been with the Department of Applied Mathematics and Computer Science, Ghent University, where he is a member of the Fuzziness and Uncertainty Modelling Research Unit and is working toward his PhD degree with a thesis on fuzzy techniques in image processing and fuzzy filters for image noise removal under the promotorship of Prof. E. E. Kerre.

Etienne Kerre received his MSc degree in mathematics in 1967 and his PhD degree in mathematics in 1970 from Ghent University, where since 1984 he has been a lector, and since 1991 a full professor. He is a referee for more than 30 international scientific journals, and a member of the editorial board of international journals and conferences on fuzzy set theory. He was an honorary chair at various international conferences. In 1976 he founded the Fuzziness and Uncertainty Modelling Research Unit (FUM) and his research has since focused on the modeling of fuzziness and uncertainty and has resulted in a great number of contributions in fuzzy set theory and its various generalizations. Especially the theories of fuzzy relational calculus and of fuzzy mathematical structures owe a very great deal to him. Kerpe has also been a promotor of 20 PhDs on fuzzy set theory. His current research interests include fuzzy and intuitionistic fuzzy relations, fuzzy topology, and fuzzy image processing. He has authored or coauthored 11 books and more than 300 papers that appeared in international refereed journals and conference proceedings.