Video Denoising by Fuzzy Motion and Detail Adaptive Averaging

Tom Mélange,* Mike Nachtegael, and Etienne E. Kerre

Ghent University,

Department of Applied Mathematics and Computer Science,

Fuzziness and Uncertainty Modelling Research Unit,

Krijgslaan 281 (Building S9),

9000 Gent, Belgium

†

Vladimir Zlokolica

MicronasNIT Institute, Fruskogorska 11,

21000 Novi Sad, Serbia&Montenegro

Stefan Schulte

Traficon N.V., Vlamingstraat 19,

8560 Wevelgem, Belgium

Valérie De Witte

Vlaamse Radio- en Televisieomroep,

Auguste Reyerslaan 52,

1043 Brussel, Belgium

Aleksandra Pizurica and Wilfried Philips

Ghent University,

Dept. of Telecommunications and Information Processing

(TELIN), IPI, Sint-Pietersnieuwstraat 41,

9000 Gent, Belgium

(Dated: June 13, 2008)

Abstract

A new fuzzy-rule based algorithm for the denoising of video sequences corrupted with additive Gaussian noise is presented. The proposed method constitutes a fuzzy logic based improvement of a recent detail and motion adaptive multiple class averaging filter (MCA). The method is first explained in the pixel domain for greyscale sequences and is later extended to the wavelet domain and to colour sequences. Experimental results show that the noise in digital image sequences is efficiently removed by the proposed fuzzy motion and detail adaptive video filter (FMDAF) and that the method outperforms other state-of-the-art filters of comparable complexity on different video sequences.

*Electronic address: Tom.Melange@ugent.be

†URL: http://www.fuzzy.ugent.be

I. INTRODUCTION

Very often, image sequences are corrupted with noise due to bad acquisition, transmission or recording. Some well-known noise types that may occur are e.g. impulse noise, Gaussian noise, speckle noise etc. In this paper we will concentrate on an additive white Gaussian noise model of zero mean and variance σ^2:

$$I_{n,i} = I_{o,i} + \epsilon_i, \quad i = 1, \ldots, p \tag{1}$$

where $I_{n,i}$ and $I_{o,i}$ denote the i-th pixel from the noisy and the original frame respectively, $\epsilon_i \sim N(0, \sigma^2)$ and p is the number of pixels per frame.

The goals of reducing the noise in the sequences are (i) visual improvement and (ii) improvement of further analysis or coding of the sequences.

The first filters for video denoising were single resolution filters. These were often some well-known 2D filters extended to a spatio-temporal neighbourhood. Some examples are the 3D KNN-filter [1–3] and the 3D threshold averaging filter [3, 4], which try to preserve the details by averaging only over the k nearest neighbours (KNN) and the neighbours lying within a certain distance (usually 2σ is chosen as threshold) from the given pixel value respectively. More recent extensions of these filters, that are made more adaptive to a local spatio-temporal neighbourhood are e.g. the motion and detail adaptive KNN-filter [5] and the multiple class averaging filter [6, 7]. Another well-known single resolution method is the 3D rational filter [8], where the filtered output for a pixel is determined as a rational function of the grey values in a spatio-temporal neighbourhood. Other recent single resolution filters can e.g. be found in [9, 10]. Both filters take into account pixels from neighbouring frames in the averaging, which not neccessarily are the pixels at the same spatial position, but the estimated corresponding object pixels which possibly have been displaced due to motion between frames.

Later, the wavelet transform, which has proven very effective in still image denoising [11, 12], also found its way in the denoising of videos. In [13, 14] a 3D wavelet transform is applied and the resulting coefficients are denoised by adaptive thresholding. However, most wavelet domain filters use a less complex separable 2D transform applied on each frame separately [6, 7, 15–22] and combine it with time-recursive filtering, either in the wavelet domain or in the pixel domain.

The most fundamental difference between video and image denoising is that in video applications also information from previous frames is available. When working with a delay in time even information from future frames can be used. The main difficulty in exploiting this additional info is possible motion. Some filters simply take into account pixels at corresponding positions in the previous (and future) frames only when no motion between the successive frames is detected. Such motion detection filters are for example [5–7, 15]. Other more complex filters always take into account information from the previous frames, by filtering along an estimated motion trajectory and are called motion compensated filters [9, 10, 16, 19, 21]. In [9, 10, 16] the motion is estimated in the pixel domain, while in [19, 21] the motion vectors are computed in the wavelet domain. Most available motion estimation algorithms are designed for video coding applications [23–25]. In such applications, the accuracy of the motion vectors is less important than for denoising purposes. Recently, in [20], an efficient video filtering scheme is proposed, which makes use of motion estimators from video codecs, but with additional filtering of the motion vectors and with appropriately defined reliabilities to estimated motion.

The filter in [10] only filters temporally. Usually however, the temporal filtering, which uses information from neighbouring frames, is combined with a spatial filtering. When the spatial and temporal filtering steps are performed separately, i.e., the one after the other, we speak of a separable filter [15, 16, 18–21]. In [18] e.g., the authors combine their image denoising method from [12] with a selective wavelet shrinkage method which estimates the level of noise corruption as well as the amount of motion in the image sequence. Filters that integrate spatial and temporal filtering in one step, such as [5–9, 13, 14, 22, 26], are called non-separable.

The method proposed in this paper is a fuzzy logic based improvement of the multiple class averaging filter (MCA) from [6, 7] for the denoising of greyscale image sequences corrupted with additive white Gaussian noise. Fuzzy set theory and fuzzy logic offer us a powerful tool for representing and processing human knowledge. Binary decisions are replaced by a gradual transition, which is more appropriate when dealing with complex systems. Examples that illustrate the power of fuzzy set theory in the domain of image processing are e.g. [27, 28]. The main differences between the proposed method and the filter from [6, 7] are: (i) pixels are not divided into discrete classes and dealt with based on their class index like in [6, 7], but they are treated individually, which leads to an increased

performance; (ii) the complicated heuristic construction of exponential functions to tune the pixel weights in the method of [6, 7] to the class index and to the detected motion and detail is replaced by a fuzzy rule containing linguistic variables, which represent human knowledge and which are more natural to work with and to understand. The use of fuzzy logic also provides a more theoretical base; (iii) in the wavelet-based extension of the method, we opt for an additional time-recursive averaging instead of a filtering of the low-frequency band ; and (iv) the fuzzy rule used in our method is easy to extend and to include new information in future work.

In this paper we also extend the proposed method to the processing of colour image sequences. We present a new vector based extension of the proposed greyscale method using the $L^*a^*b^*$-transform in combination with a 3D extension of the colour restorating second subfilter from [29].

Experimental results show that our method outperforms other state-of-the-art filters of a comparable complexity.

The paper is structured as follows: Our algorithm for the denoising of greyscale image sequences is first explained in the pixel domain in Section II and extended to the wavelet domain in Section III. In Section IV we discuss the processing of colour video. Section V handles the choice of the parameter values. Finally, experimental results and conclusions are presented in Section VI and Section VII respectively.

II. PIXEL-BASED SPATIO-TEMPORAL FILTER FOR GRAYSCALE VIDEO

In this section, we improve the multiple class averaging filter (MCA) from [6, 7] in the pixel domain by incorporating fuzzy logic. The ideas behind the filter are the following: (i) to avoid spatio-temporal blur, one should only take into account neighbouring pixels from the current frame in case of detected motion; (ii) to preserve the details in the frame content, the filtering should be less strong when large spatial activity (e.g. a large variance) is detected in the current filtering window. As a consequence more noise will be left, but large spatial activity corresponds to high spatial frequencies and for these the eye is less sensitive [30]. In the case of homogeneous areas, strong filtering should be performed to remove as much noise as possible.

The general filtering framework used in the proposed method is presented in Subsec-

tion II A. Additionally the crucial weight determination step, which is the main novelty of our greyscale method compared to the MCA filter, is explained in Subsection II B. In the proposed method we determine the weights in the filtering window by the use of fuzzy sets and fuzzy logic instead of a heuristic construction with exponential functions as it is the case in the MCA filter. Subsection II C finally, discusses some complexity notes.

A. The General Filtering Framework

In this Subsection, the filtering framework used in both the MCA and the proposed filter is explained. In the following, a noisy input image pixel and the corresponding filtered pixel value are denoted by respectively $I_n(x, y, t)$ and $I_f(x, y, t)$, where (x, y) indicates the spatial location and t stands for the temporal location.

The filtering window used in the framework is a $3 \times 3 \times 2$ sliding window, consisting of 3×3 pixels in the current frame and 3×3 pixels in the previous frame. As introduced in [6, 7] we will use the terms *current window* and *previous window* for the window pixels contained in respectively the current and the previous frame (Fig. 1). This window is moved

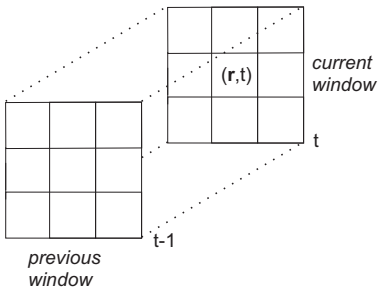

FIG. 1: The $3 \times 3 \times 2$ filtering window consisting of the previous and the current window.

through each frame from top left to bottom right, each time filtering the central pixel by averaging the noise. The position of this central pixel in the filtering window is denoted by $(\mathbf{r}, t)$ where $\mathbf{r} = (x, y)$ stands for the spatial position and t for the temporal position. An arbitrary position in the $3 \times 3 \times 2$ window (this may also be the central pixel position) is denoted by $(\mathbf{r'}, t')$, with $\mathbf{r'} = (x + k, y + l)$ $(-1 \leq k, l \leq 1)$ and $t' = t$ or $t' = t - 1$.

The output of the proposed filter for the central pixel in the window is finally determined

as a weighted average (with adaptive weights) of the pixel values in the $3 \times 3 \times 2$ window:

$$I_f(\mathbf{r},\mathbf{t}) = \frac{\sum_{\mathbf{r}'} \sum_{t'=t-1}^{t} W(\mathbf{r}', t', \mathbf{r}, t) I_n(\mathbf{r}', t')}{\sum_{\mathbf{r}'} \sum_{t'=t-1}^{t} W(\mathbf{r}', t', \mathbf{r}, t)}. \tag{2}$$

B. Weight Determination

In this subsection, we focus on the fundamental step in the filtering framework, namely the determination of the weights. To make the method motion and detail adaptive, we adopt the difference value $\Delta(\mathbf{r}', t', \mathbf{r}, t)$, the detail value $d(\mathbf{r}, t)$ and the motion value $m(\mathbf{r}, t)$ from [6, 7]:

(i) The absolute greyscale difference between the two pixel positions $(\mathbf{r}, t)$ and $(\mathbf{r}', t')$ is denoted by:

$$\Delta(\mathbf{r}', t', \mathbf{r}, t) = |I_n(\mathbf{r}', t') - I_n(\mathbf{r}, t)|. \tag{3}$$

(ii) The function $d(\mathbf{r}, t)$ indicating the local amount of detail is calculated as the standard deviation in the current window:

$$I_{av}(\mathbf{r}, t) = \frac{1}{9} \sum_{\mathbf{r}'} I_n(\mathbf{r}', t) \ , \tag{4}$$

$$d(\mathbf{r}, t) = \Big(\frac{1}{9} \sum_{\mathbf{r}'} \big(I_n(\mathbf{r}', t) - I_{av}(\mathbf{r}, t) \big)^2 \Big)^{\frac{1}{2}} \ . \tag{5}$$

(iii) The motion indicator $m(\mathbf{r}, t)$ finally, is defined as the absolute difference between the average grey value in the current window and the average grey value in the previous window:

$$\begin{aligned} m(\mathbf{r}, t) &= |I_{av}(\mathbf{r}, t) - I_{av}(\mathbf{r}, t-1)| \\ &= |\frac{1}{9} \sum_{\mathbf{r}'} I_n(\mathbf{r}', t) - \frac{1}{9} \sum_{\mathbf{r}'} I_n(\mathbf{r}', t-1)|. \end{aligned} \tag{6}$$

1. MCA Filter

In the MCA filter [6, 7], the pixels are classified into four discrete index classes, depending on the $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$ value:

$$i(\mathbf{r'}, t', \mathbf{r}, t) = \begin{cases} 0, & \Delta(\mathbf{r'}, t', \mathbf{r}, t) \leq k\sigma_n \\ 1, & k\sigma_n < \Delta(\mathbf{r'}, t', \mathbf{r}, t) \leq 2k\sigma_n \\ 2, & 2k\sigma_n < \Delta(\mathbf{r'}, t', \mathbf{r}, t) \leq 3k\sigma_n \\ 3, & 3k\sigma_n < \Delta(\mathbf{r'}, t', \mathbf{r}, t) \end{cases} \tag{7}$$

where σ_n represents the standard deviation of the Gaussian noise and k is a parameter. When details are detected in a region, higher weights are assigned to pixels which are similar to the pixel being filtered (i.e. pixels from the lower index classes, which have smallest $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$ values) to preserve these details. In homogeneous regions however, the difference in weight compared to pixels from the higher index classes will be smaller and strong filtering is performed. This is done by determining the weights by a heuristic composition of exponential functions that is inversely proportional to the amout of detail and motion and the class index. In [6] the weights for the pixels in the window are defined as:

$$W(\mathbf{r'}, t', \mathbf{r}, t) = \begin{cases} exp\left(\frac{-i(\mathbf{r'}, t', \mathbf{r}, t)}{\eta(d(\mathbf{r}, t))\sigma_n}\right)\beta(m(\mathbf{r}, t), t'), & i = 0, 1, 2 \\ 0, & i = 3 \end{cases} \tag{8}$$

where the function

$$\eta(d) = K_1 exp(-K_2 d) + K_3 exp(-K_4 d), \tag{9}$$

is used to determine the slope of the exponential function in (8) and K_1, K_2, K_3 and K_4 are parameters. The function $\beta(m(\mathbf{r}, t), t')$ in (8) is chosen to limit the contribution (decreasing the weight) of the pixels from the previous window in case of motion:

$$\beta(m(\mathbf{r}, t), t') = \begin{cases} 1, & t' = t \\ exp(-\gamma m(\mathbf{r}, t)), & t' = t - 1 \end{cases} \tag{10}$$

In this equation, the parameter γ is used to control the sensitivity of the motion detector. In [7] the function $\eta(d)$ is omitted and the weights are then defined as:

$$W(\mathbf{r'}, t', \mathbf{r}, t) = \begin{cases} exp\left(\frac{-i(\mathbf{r'}, t', \mathbf{r}, t)d(\mathbf{r}, t)}{K_d\sigma_n}\right)\beta(m(\mathbf{r}, t), t'), & i = 0, 1, 2 \\ 0, & i = 3 \end{cases} \tag{11}$$

where K_d is a parameter.

2. Proposed Filter

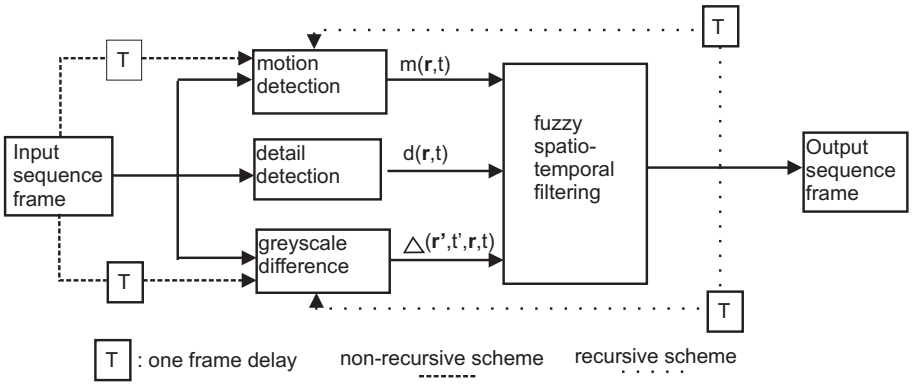

FIG. 2: The general filtering scheme of the proposed filter.

In our fuzzy motion and detail adaptive video filter, we use the above introduced filtering framework and the values $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$, $m(\mathbf{r}, t)$ and $d(\mathbf{r}, t)$ (Fig. 2). In contrast to the MCA filter we no longer use discrete index classes to express the similarity of a pixel to the central window pixel. Also our determination of the weights in (2) differs from the strategy used in [6, 7]. The artificial construction of exponential functions in the MCA method is replaced by a more natural fuzzy logic framework with linguistic variables.

The four index classes are replaced by one fuzzy set [31] "large difference" for the values $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$. A fuzzy set C in a universe Y is characterized by a $Y \to [0, 1]$ mapping μ_C, which associates with every element y in Y a degree of membership $\mu_C(y)$ of y in the fuzzy set C. For example, if a difference $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$ has a membership degree one in the fuzzy set "large difference", then this means that this difference is large for sure. A membership degree equal to zero would express the certainty that the difference is not large. Membership degrees between zero and one mean that we can neither say that the difference is definitely large, nor that the difference would not be large. The membership degree is however an indication of whether the difference is large rather than small. So, a pixel $I_n(\mathbf{r'}, t')$ that would belong to a low index class in the MCA filter now corresponds to a small membership degree of the value $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$ in the fuzzy set "large difference". We will use a linguistic variable "large" not only for the difference $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$, but also for the motion value $m(\mathbf{r}, t)$,

for the detail value $d(\mathbf{r}, t)$ and introduce the fuzzy sets "large motion", "large detail" and "large weight". We will furhter also use a linguistic variable "reliable" to indicate whether a given neighbourhood pixel is reliable to be used in the filtering of the central window pixel, and represent it by the fuzzy set "reliable neighbourhood pixel".

In the following the notations μ_Δ, μ_d and μ_m are used to denote the membership functions characterizing respectively the fuzzy sets (i) large difference, (ii) large detail and (iii) large motion. For the sake of simplicity and computational reasons triangular functions are used, as shown in Fig. 3. As can be seen in Fig. 3, the membership functions are completely

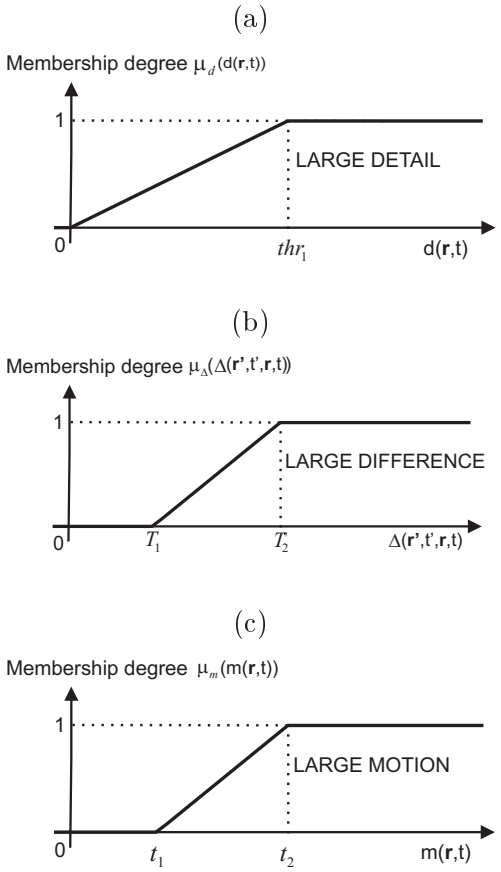

FIG. 3: (a) The membership function μ_d for the fuzzy set "large detail", (b) The membership function μ_Δ for the fuzzy set "large difference" and (c) The membership function μ_m for the fuzzy set "large motion".

determined by the parameters thr_1, T_1, T_2, t_1 and t_2.

Using the introduced fuzzy sets for the crucial weight determining step, we replace the heuristic combination of exponential functions in the original MCA method by a more nat-

ural fuzzy logic framework with linguistic variables. The weight $W(\mathbf{r'}, t', \mathbf{r}, t)$ for the pixel at position ($\mathbf{r'}$,t') is now defined as the degree to which it is reliable to be used in the filtering of the central window pixel, i.e., its membership degree in the fuzzy set "reliable neighbourhood pixel", which is the activation degree of the Fuzzy Rule 1 or 2 depending on whether $t' = t$ or $t' = t - 1$. The general form of a fuzzy rule is "IF A THEN B", where the premise A (also called the antecedent) and the consequent B are (collections of) propositions containing linguistic variables.

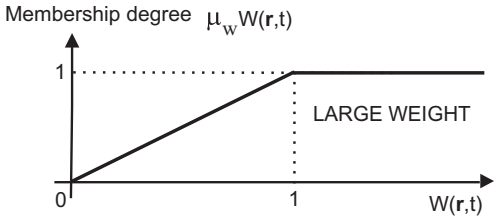

FIG. 4: The membership function μ_w for the fuzzy set "large weight".

Fuzzy Rule 1 *Assigning the membership degree in the fuzzy set "reliable neighbourhood pixel" of the pixel at spatial position* $\mathbf{r'}$ *in the current frame* ($t' = t$) *of the window with central pixel position* ($\mathbf{r}$,t)*:*

IF(*the detail value* $d(\mathbf{r}, t)$ *is* large AND *the difference* $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$ *is*

not large) OR (*the detail value* $d(\mathbf{r}, t)$ *is* not large)

THEN *the pixel at position* ($\mathbf{r'}$,t') *is a* reliable *neighbourhood pixel for the filtering of the central window pixel.*

Fuzzy Rule 2 *Assigning the membership degree in the fuzzy set "reliable neighbourhood pixel" of the pixel at spatial position* $\mathbf{r'}$ *in the previous frame* ($t' = t - 1$) *of the window with central pixel position* ($\mathbf{r}$,t)*:*

IF((*the detail value* $d(\mathbf{r}, t)$ *is* large AND *the difference* $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$ *is*

not large) OR (*the detail value* $d(\mathbf{r}, t)$ *is* not large))

AND *the motion value* $m(\mathbf{r}, t)$ *is* not large

THEN *the pixel at position* ($\mathbf{r'}$,t') *is a* reliable *neighbourhood pixel for the filtering of the central window pixel.*

The AND and OR operators used in Fuzzy Rules 1 and 2 correspond to respectively intersections and unions of two fuzzy sets. The intersection of two fuzzy sets A and B in a universe Y is specified by a mapping T that maps the membership degrees of an element in the fuzzy sets A and B onto a membership degree in the fuzzy set $A \cap B$: $\mu_{(A \cap B)}(y) = T(\mu_A(y), \mu_B(y)), \forall y \in Y$. Analogously, the membership degree of an element in the union of A and B is obtained from the membership degrees in A and B through the help of a mapping S: $\mu_{(A \cup B)}(y) = S(\mu_A(y), \mu_B(y)), \forall y \in Y$. In fuzzy logic for the mapping T a triangular norm [33] is used, while for the mapping S a triangular conorm [33] is used. Some well-known triangular norms together with their dual conorms can be found in Table I. From all possible triangular conorms the strong conorm is the largest and the maximum conorm is the smallest. We have chosen for a triangular norm and its dual conorm which is situated in between those two extremes, i.e., the algebraic product and the probabilistic sum, respectively. As demonstrated in Subsection VI C, we see a comparable performance when using other norms and conorms.

TABLE I: Some well-known triangular norms and triangular conorms.

Triangular norms	
minimum	$\min(x, y)$
algebraic product	$x \cdot y$
weak	$\begin{cases} \min(x, y) & \text{if } \max(x, y) = 1 \\ 0 & \text{otherwise} \end{cases}$
Łukasiewicz	$\max(0, x + y - 1)$
Triangular conorms	
maximum	$\max(x, y)$
probabilistic sum	$x + y - x \cdot y$
strong	$\begin{cases} \max(x, y) & \text{if } \min(x, y) = 0 \\ 1 & \text{otherwise} \end{cases}$
Łukasiewicz	$\min(1, x + y)$

The fuzzy rules further also contain NOT operators, corresponding to the complement

of a fuzzy set A. In fuzzy logic, the complement of a fuzzy set is specified by an involutive negator [33]. For the results in this paper, we have used the well-known standard negator $N(x) = 1 - x$, $\forall x \in [0, 1]$. The membership degree of an element in the complement of a fuzzy set A in Y is then given by: $\mu_{(co(A))}(y) = N(\mu_A(y)) = 1 - \mu_A(y)$, $\forall y \in Y$.

Take for example Fuzzy Rule 1. This rule has an activation degree (corresponding to the membership degree in the fuzzy set "reliable neighbourhood pixel" and thus the weight $W(\mathbf{r'}, t', \mathbf{r}, t)$ in (2) for the pixel in the sliding window at position $(\mathbf{r'}, t')$) equal to:

$$\alpha_1 \cdot (1 - \alpha_2) + (1 - \alpha_1) - \alpha_1 \cdot (1 - \alpha_2) \cdot (1 - \alpha_1), \tag{12}$$

with $\alpha_1 = \mu_d(d(\mathbf{r}, t))$ and $\alpha_2 = \mu_\Delta(\Delta(\mathbf{r'}, t', \mathbf{r}, t))$.

Notice that it is impossible that all weights in (2) are equal to zero. In the above expression either α_1 or $1 - \alpha_1$ is always greater than zero ($\alpha_1 \in [0, 1]$), and for the central pixel position $\mathbf{r}$, we always have that $\alpha_2 = 0$ (see expression (3) and Fig. 3 (b)).

The proposed fuzzy rules are very natural to work with since they directly express the underlying ideas put in a formal framework: (i) When large spatial activity is detected, one should filter less to preserve the details. This means that the neighbouring pixels that are assigned a considerable weight in (2), should be similar to the central pixel in the filtering window ($d(\mathbf{r}, t)$ is large AND $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$ is not large). In the opposite case (OR), i.e., in homogeneous areas ($d(\mathbf{r}, t)$ is not large) no extra conditions should be imposed on the neighbouring pixels. All pixels should get a considerable weight to perform strong smoothing. (ii) When motion is detected between the current and the previous window, only pixels from the current frame should be taken into account in the averaging. This means that pixels from the previous frame only should get a considerable weight when the motion detector yields a low value ($m(\mathbf{r}, t)$ is not large) (corresponding to the second (AND) in Fuzzy Rule 2).

Apart from being a formal representation of the ideas, the fuzzy rules also produce the desired result. In the case of spatio-temporal structures, the detail and motion value will be large and only for neighbouring pixels with a small difference in greyscale value (relative to the central pixel in the filtering window), the Fuzzy Rules 1 and 2 will have a considerable activation degree. In this way fine spatio-temporal details are preserved at the expense of some noise reduction.

In a spatio-temporal uniform area, the detail and motion values will not be large. So even for neighbouring pixels with a large difference in greyscale value (relative to the central

pixel), the Fuzzy Rules 1 and 2 will have a considerable activation degree. Hence, because of the many considerable weights in (2), strong filtering is performed.

Finally, we also propose a recursive scheme of the fuzzy motion and detail adaptive video filter. In this scheme, we always use the filtered value $I_f(\mathbf{r'}, t-1)$ for the neighbouring pixels in the already filtered previous frame. For pixels in the current frame, the noisy values $I_n(\mathbf{r'}, t-1)$ are used, except for the determination of $\Delta(\mathbf{r'}, t', \mathbf{r}, t)$, where the filtered value is used when already available (i.e., for pixels that have been filtered already in a previous step). In this way, we get a better estimate of whether the pixel at position ($\mathbf{r'}$,t') belongs to the same object as the pixel at position $(\mathbf{r}, t)$ or not.

C. Some Complexity Notes

It is clear that the complexity of the proposed filter is linear in terms of the number of pixels in a frame. Every pixel is filtered by averaging a constant number of neighbourhood pixels, which are all assigned a weight using a constant number of operations. The calculation of the activation degree of the used fuzzy rules has a low complexity. The activation degree of Fuzzy Rule 1 is given in expression (12). For Fuzzy Rule 2, an extra multiplication with $(1-\alpha_3)$ $(\alpha_3 = \mu_m(m(\mathbf{r}, t)))$ is needed. To calculate the activation degree of Fuzzy Rule 1, 3 multiplications, 2 sums and 3 subtractions are performed. For the activation degree of Fuzzy Rule 2 an extra subtraction and product are required. For the MCA filter, the calculation of the weight in expression (8) requires 7 multiplications, one division, and the calculation of 3 exponential functions and 4 opposites. The alternative in expression (11) can be computed by 4 multiplications, one division and the calculation of 2 exponential functions and 2 opposites. The use of fuzzy logic in the weight calculation is thus not more complex. The proposed individual treatment of the pixels, however, requires the weight calculation for each individual pixel. In the MCA filter, weights are only calculated for the different index classes, which results in a little lower complexity.

III. WAVELET-BASED SPATIO-TEMPORAL FILTER WITH ADDITIONAL PIXEL-BASED TIME-RECURSIVE AVERAGING FOR GRAYSCALE VIDEO

In this section our method is extended to the wavelet domain. The procedure is the following: each processed frame is first decomposed using the 2D wavelet transform [34]. Next, an adapted version of the proposed method from Section II is applied on each of the resulting wavelet bands separately. Finally, the inverse wavelet transform is applied, followed by an additional time-recursive averaging in the pixel domain (see Fig. 5).

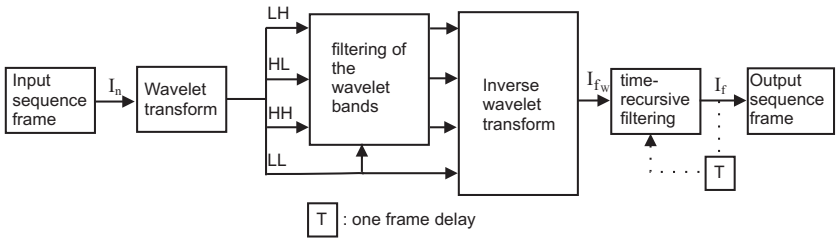

FIG. 5: The filtering scheme for the proposed wavelet domain method.

A. Basic Notions

The wavelet transform of an image results in a representation that is very useful for image denoising. The transform compacts image details (such as edges and texture) into a small number of spatially clustered large coefficients, while small coefficients correspond to homogeneous regions in the original image.

We use the notation $y_{s,d}(\mathbf{r}, t)$ for the wavelet coefficient at resolution scale s, orientation d and spatial position $\mathbf{r}$ of the frame with temporal position t. For the results in this paper, we have opted for a wavelet decomposition with three orientation subbands, leading to three detail images at each scale, characterized by horizontal ($d = LH$), vertical ($d = HL$) and diagonal ($d = HH$) directions and a low-frequency band (denoted by LL). Whenever there can be no confusion, we omit the indices s and d.

Due to the linearity of the wavelet transform, additive noise in the pixel domain remains additive after the transformation as well, resulting in:

$$y(\mathbf{r}, t) = \beta(\mathbf{r}, t) + \epsilon(\mathbf{r}, t),$$

where $y(\mathbf{r},t)$ and $\beta(\mathbf{r},t)$ are respectively the noisy and the noise-free wavelet coefficients and $\epsilon(\mathbf{r},t)$ is the corresponding noise component.

B. Fuzzy Motion and Detail Adaptive Averaging in the Wavelet Domain

The proposed method is now extended to the wavelet domain. Large differences in grey value in the pixel domain indicate the occurence of an edge. To preserve the edges, pixels with a large difference in grey value, relative to the pixel being filtered in the current step, should not be taken into account in the averaging. Only pixels from the same object, i.e., belonging to the same side of the edge, should be averaged and are expected to have a similar grey value. In the wavelet domain, edges result in large coefficients. So to preserve the edges, only the large coefficients, corresponding to these edges, should be averaged to filter out the noise. Small coefficients should get small weights in this case, and vice versa for homogeneous areas. This also holds for wavelet coefficients in the previous window. When there is no motion, the wavelet coefficients corresponding to the same edge in the previous frame are expected to be of a similar size. Hence, similar values should result in large weights and large differences in small weights.

Because the region of wavelet coefficients that are influenced by a given pixel value expands with increasing scale, an averaging scheme become less and less efficient for higher scales. Therefore we have used only two scales in the wavelet decomposition, which is insufficient to remove all the noise. To overcome this problem, in [6, 7], also the low-frequency band is filtered to obtain a better noise removal. In this paper, we choose instead for an additional time-recursive filtering in the pixel domain like in [15], but in a more adaptive fuzzy logic based way.

1. *Filtering of the Wavelet Bands*

The filtering of the wavelet bands is adapted in an analogous way as in [6, 7]:

- We adopt the corresponding definition for the detail value $d(\mathbf{r},t)$ from [6, 7]:

$$d(\mathbf{r},t) = \Big(\sum_{\mathbf{r'}} y^2_{s,d}(\mathbf{r'},t) \Big)^{\frac{1}{2}}. \tag{13}$$

- For all detail bands the same motion indicator value is used, which is computed on the low-frequency band. This motion value is defined as the absolute difference between the central coefficient value in the current window and in the previous window of the low-frequency band.

- The parameters that define the membership functions μ_Δ, μ_d and μ_m in Fig. 3 need to be adapted to the specific detail band.

Since $m(\mathbf{r},t)$, $d(\mathbf{r},t)$ and $\Delta(\mathbf{r'},t',\mathbf{r},t)$ are all three defined, Fuzzy Rules 1 and 2 can still be used to determine the weights in (2). The only difference is that we are now working with wavelet coefficients instead of pixel values.

2. Additional Time-Recursive Filter in the Pixel Domain

Let I_{f_W} and I_f respectively denote the frame after the filtering of the wavelet bands and the inverse wavelet transform and the frame after the additional time-recursive filtering (see Fig. 5).

First, the absolute difference between the pixels in the current frame after the filtering of the wavelet bands and the pixel at the corresponding position in the previous frame, which has already been processed by the additional time-recursive filter, is computed:

$$TD(\mathbf{r},t) = |I_{f_W}(\mathbf{r},t) - I_f(\mathbf{r},t-1)|. \tag{14}$$

For each difference, its membership degree $\mu_{TD}(TD(\mathbf{r},t))$ in the fuzzy set "large temporal difference" is then calculated. The membership function μ_{TD} of this fuzzy set is depicted in Fig. 6.

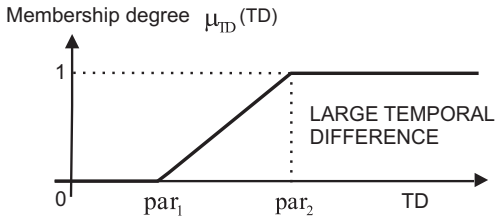

FIG. 6: The membership function μ_{TD} for the fuzzy set "large temporal difference".

The final output of the additional time-recursive filter is given by

$$I_f(\mathbf{r},t) = \frac{1-\mu_{TD}(TD(\mathbf{r},t))}{2} I_f(\mathbf{r},t-1) + \frac{1+\mu_{TD}(TD(\mathbf{r},t))}{2} I_{f_W}(\mathbf{r},t), \tag{15}$$

where the contribution of $I_f(\mathbf{r}, t-1)$ is limited to a maximum of $\frac{1}{2}$ to prevent noise propagation in time.

IV. PIXEL-BASED SPATIO-TEMPORAL FILTER FOR COLOUR VIDEO

In this Section we propose a new scheme to handle colour image sequences. As in most image processing applications, we assume that the colour frames are represented in the RGB colour space. The different colours in this RGB space are obtained by adding the three colours red, green and blue together in different proportions. As a consequence, an input frame of a colour video can be represented by a 2D matrix of 3D vectors, containing the amount of red ($I(x, y, t, 1)$), green ($I(x, y, t, 2)$) and blue ($I(x, y, t, 3)$) for a given pixel $I(x, y, t)$ in the 2D matrix.

A first straightforward way to process colour video with the proposed method is to process each of the colour bands (R, G and B) separately. In this way however, the correlation between the colour channels is neglected and unwanted colour artefacts are often introduced.

The scheme that is usually applied, consists of denoising the luminance component of the YUV-transform. In this colour sequence denoising scheme, the colour frames are first converted from the RGB colour space into the YUV colour space, by a linear transformation. The Y-component in this space contains the information about the luminance of the image, while the information about the colour (hue and saturation) is encoded in the U- and V-component. The Y-component is called the luminance component and the U- and V-component together are called the chrominance components. Since the human eye is far less sensitive to spatial details in chrominance than in luminance [35], it is acceptable to only filter the luminance component. In this way, only one band is filtered instead of three. To achieve better results, a simple additional filtering of the chrominance bands (e.g. spatial averaging) can be applied. Afterwards, the inverse YUV-RGB transform is applied.

In this section we introduce a new alternative where the $L^*a^*b^*$- colour space is used. We first present a vector based extension of the proposed greyscale method (Section II) in Subsection IV A (first subfilter) and then combine it with a 3D extension of the color restoration second subfilter from [29] in Subsection IV B (second subfilter) (see Fig 7).

We assume that all three colour bands in the RGB colour space are contaminated with white Gaussian noise with zero mean and the same standard deviation.

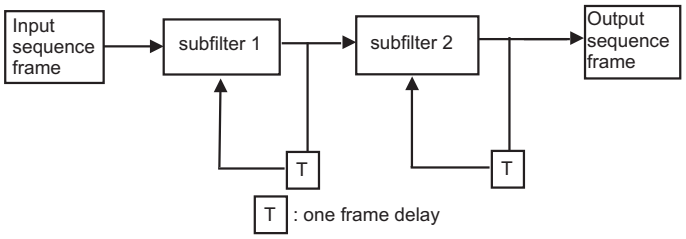

FIG. 7: The filtering scheme of the proposed colour filter.

A. First Subfilter

Since the algorithm proposed in Section II makes use of the absolute difference between grey values, for colour video it makes sense to work in a colour space in which the measured distance between colours roughly corresponds to the difference in colour as it is observed by the human eye. This is the case in the linear $L^*a^*b^*$ colour model. In the proposed colour extension, each processed frame is first transformed into the $L^*a^*b^*$ colour space. Subsequently, the transformed frame is filtered with the adapted algorithm as explained below. Finally, the filtered frame is retransformed to the RGB colour space. For the transformation between the RGB and $L^*a^*b^*$ colour spaces, the XYZ colour space is used as an intermediate step. For more information on colour spaces and their use in image processing, we refer to [36].

In the following the $L^*a^*b^*$-transform of the RGB-vector at pixel position $I_n(\mathbf{r},t)$ is denoted by $I_{n,L^*a^*b^*}(\mathbf{r},t)$, while the L^*-, a^*- and b^*-component of this vector are denoted by $I_{n,L^*}(\mathbf{r},t)$, $I_{n,a^*}(\mathbf{r},t)$ and $I_{n,b^*}(\mathbf{r},t)$ respectively.

1. *Adaptation of* $m(\boldsymbol{r},t)$, $d(\boldsymbol{r},t)$ *and* $\Delta(\boldsymbol{r'},t',\boldsymbol{r},t)$

The motion value $m(\mathbf{r},t)$ for this vector based method is determined as the Euclidian distance between the $L^*a^*b^*$-transforms of the vectors at the central pixel position of the current and the previous window:

$$\begin{aligned} m(\mathbf{r},t) &= \|I_{n,L^*a^*b^*}(\mathbf{r},t) - I_{f_1,L^*a^*b^*}(\mathbf{r},t-1)\|_2 \\ &= \Big(\big(I_{n,L^*}(\mathbf{r},t) - I_{f_1,L^*}(\mathbf{r},t-1)\big)^2 \\ &\quad +\big(I_{n,a^*}(\mathbf{r},t) - I_{f_1,a^*}(\mathbf{r},t-1)\big)^2 \\ &\quad +\big(I_{n,b^*}(\mathbf{r},t) - I_{f_1,b^*}(\mathbf{r},t-1)\big)^2\Big)^{\frac{1}{2}}, \end{aligned} \tag{16}$$

where I_{f_1} denotes the output of this first subfilter.

For the adaptation of the detail value $d(\mathbf{r},t)$, we first calculate the arithmetic mean in the current window of each of the components in the $L^*a^*b^*$-colour space:

$$\overline{L}(\mathbf{r},t) = \frac{1}{9}\sum_{\mathbf{r'}} I_{n,L^*}(\mathbf{r'},t), \tag{17}$$

$$\overline{a}(\mathbf{r},t) = \frac{1}{9}\sum_{\mathbf{r'}} I_{n,a^*}(\mathbf{r'},t), \tag{18}$$

$$\overline{b}(\mathbf{r},t) = \frac{1}{9}\sum_{\mathbf{r'}} I_{n,b^*}(\mathbf{r'},t). \tag{19}$$

The detail value itself is then defined as:

$$d(\mathbf{r},t) = \Big(\frac{1}{9}\sum_{\mathbf{r'}} \big\| I_{n,L^*a^*b^*}(\mathbf{r'},t) - (\overline{L}(\mathbf{r},t),\overline{a}(\mathbf{r},t),\overline{b}(\mathbf{r},t))\big\|_2^2\Big)^{\frac{1}{2}}. \tag{20}$$

Finally, the adapted $\Delta(\mathbf{r'},t',\mathbf{r},t)$-value is given by

$$\begin{aligned}\Delta(\mathbf{r'},t',\mathbf{r},t) &= \|I_{n,L^*a^*b^*}(\mathbf{r'},t') - I_{n,L^*a^*b^*}(\mathbf{r},t)\|_2 \\ &= \Big(\big(I_{n,L^*}(\mathbf{r'},t') - I_{n,L^*}(\mathbf{r},t)\big)^2 \\ &\quad +\big(I_{n,a^*}(\mathbf{r'},t') - I_{n,a^*}(\mathbf{r},t)\big)^2 \\ &\quad +\big(I_{n,b^*}(\mathbf{r'},t') - I_{n,b^*}(\mathbf{r},t)\big)^2\Big)^{\frac{1}{2}},\end{aligned} \tag{21}$$

for pixels in the current frame ($t' = t$), and by

$$\Delta(\mathbf{r'},t',\mathbf{r},t) = \|I_{f_1,L^*a^*b^*}(\mathbf{r'},t') - I_{n,L^*a^*b^*}(\mathbf{r},t)\|_2 \tag{22}$$

for pixels in the previous frame ($t' = t-1$).

2. Determination of the Weights

With the use of the above introduced adaptations of $m(\mathbf{r},t)$, $d(\mathbf{r},t)$ and $\Delta(\mathbf{r'},t',\mathbf{r},t)$, the weights $W(\mathbf{r'},t',\mathbf{r},t)$ in the weighted sum (2) can still be determined by the Fuzzy Rules 1 and 2. We only need to adapt the parameters thr_1, T_1, T_2, t_1 and t_2 of the membership functions μ_d, μ_m and μ_Δ to this new colour space.

B. Second Subfilter

When we consider the colour pixels as vectors, they are affected by the noise in three different dimensions, instead of in one when only considering one colour band. As a consequence, less similar neighbours can be found to average out the noise in the 3D case than in

the 1D case, and sometimes even not enough. To overcome this problem, the first subfilter is combined with a 3D extension of the colour restorating second subfilter from [29]. The central pixel in the window is estimated by combining local differences in a spatio-temporal neighbourhood, computed for the red, green and blue component each separately.

1. Local Differences and Correction Terms

Similar to the first subfilter, a $3 \times 3 \times 2$ sliding window (Fig. 1) is used. In each step the central pixel in this window, at position $(\mathbf{r}, t)$ in the image sequence, is filtered. For each pixel in the sliding window, local differences (gradients) in the three colour bands (each separately) are calculated. The differences in the red, green and blue neighbourhoods are respectively denoted by LD_1, LD_2 and LD_3. For pixels in the window belonging to the current frame, the output of the first subfilter, denoted by I_{f_1}, is used:

$$LD_i(\mathbf{r'}, t) \;=\; I_{f_1}(\mathbf{r'}, t, i) \;-\; I_{f_1}(\mathbf{r}, t, i), \tag{23}$$

with $i = 1, 2, 3$. For pixels in the window belonging to the previous frame, the already present output of the second subfilter, denoted by I_f, is used:

$$LD_i(\mathbf{r'}, t-1) \;=\; I_f(\mathbf{r'}, t-1, i) \;-\; I_{f_1}(\mathbf{r}, t, i), \tag{24}$$

again with $i = 1, 2, 3$.

Next, for each position in the window one correction term is determined using the calculated local differences. This correction term is defined as the arithmetic average of the local difference in the red, green and blue component at the given position:

$$\epsilon(\mathbf{r'}, t') \;=\; \frac{1}{3}\Big(LD_R(\mathbf{r'}, t') \;+\; LD_G(\mathbf{r'}, t') \;+\; LD_B(\mathbf{r'}, t')\Big). \tag{25}$$

2. Output of the second subfilter

Finally the output of the second subfilter for the central pixel in the current window is determined as follows:

$$\begin{aligned} I_f(\mathbf{r}, t, i) \;=\; & \frac{\sum_{\mathbf{r'}} \Big(I_{f_1}(\mathbf{r'}, t, i) - \epsilon(\mathbf{r'}, t)\Big)}{18} + \\ & \frac{\sum_{\mathbf{r'}} \Big(I_f(\mathbf{r'}, t-1, i) - \epsilon(\mathbf{r'}, t-1)\Big)}{18}, \end{aligned} \tag{26}$$

where $\epsilon(\mathbf{r'}, t')$ is the correction term for the neighbouring pixel at position $(\mathbf{r'}, t')$ and $i = 1, 2, 3$ (for respectively the red, green and blue colour band).

V. PARAMETER SELECTION

As mentioned earlier, the membership functions in Fig. 3 and 6 are completely determined by their respective parameters. These parameter values have been experimentally optimized using the "Salesman", "Trevor", "Tennis" and "Flower Garden" sequences, which all have their own characteristics. The "Salesman" sequence represents a standard sequence with moderate detail (shelfs, books,...) and moderate motion (person). The "Trevor" sequence contains very fast motion (moving arms). In the "Tennis" sequence we deal with a zooming camera and a detailed background (wall). The "Flower garden" sequence finally, combines very detailed regions (flower field) with homogeneous regions (sky).

The parameters have been optimized in the following way. The proposed method was applied on each of the above sequences, for the different noise levels $\sigma_n = 5, 10, 15, 20, 25$ with parameters varying over the range of possible values. After plotting the optimal parameter values (in terms of PSNR) for the different sequences and noise levels, a linear relationship was found between these optimal parameter values and the noise level. Therefore the parameters have been determined by the best fit through the observations. As an illustration, the optimal values for the parameter T_2 of the proposed pixel domain method together with the best fitting line through these points are depicted in Fig. 8. The parameters are thus linearly dependent of the noise level. For the results in this paper, we assume a known standard deviation of the noise. In most practical cases however, the standard deviation σ_n is not known and should be estimated. A common used noise estimation method is the wavelet domain median absolute deviation (MAD) estimator of Donoho and Johnstone [32].

The optimized parameter values that determine the membership functions used in the pixel domain method are given in Table II.

Table III presents the optimized thr_1, T_1 and T_2 values for the different waveletbands in the wavelet domain method. For the membership function μ_m, the parameters are determined as $t_1 = 3.22\sigma_n + 1.5667$ and $t_2 = 36.7667\sigma_n + 16.5$. The optimized parameters for the additional time-recursive filtering are given by repectively $par_1 = 0.555\sigma_n - 0.725$ and $par_2 = 1.36\sigma_n + 5.1$.

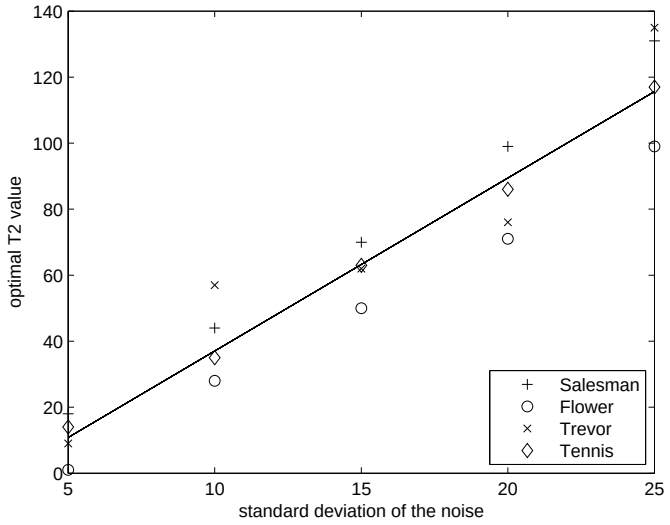

FIG. 8: Optimal value for the parameter T_2 of the proposed pixel domain method.

TABLE II: Optimized parameter values for the pixel domain method.

parameter	optimal value
thr_1	$1.36\sigma_n + 1.2$
T_1	$0.79\sigma_n + 0.25$
T_2	$5.24\sigma_n - 15.35$
t_1	$0.465\sigma_n - 0.625$
t_2	$1.795\sigma_n + 3.275$

For the vector based colour extension of the method, finally, the parameters can be found in Table IV.

VI. EXPERIMENTAL RESULTS

In this section we will show some experimental results. For the experiments, our wavelet domain algorithm has been implemented with a non-decimated wavelet transform (which is known to give better denoising results than the decimated one) using the Haar-wavelet. As mentioned before (Subsection III B) we have used only two levels in the wavelet decomposition.

TABLE III: Optimized thr_1, T_1 and T_2 values for the different detail bands.

Band	thr_1	T_1	T_2
LH_1	$5.5733\sigma_n - 14.2667$	$0.8867\sigma_n - 1.9667$	$2.94\sigma_n + 2.9$
HL_1	$5.5733\sigma_n - 14.2667$	$0.8867\sigma_n - 1.9667$	$2.94\sigma_n + 2.9$
HH_1	$46.6267\sigma_n - 243.0667$	$0.8867\sigma_n - 1.9667$	$2.94\sigma_n + 2.9$
LH_2	$2.7533\sigma_n - 1.3$	$2.7067\sigma_n - 8.2667$	$2.8867\sigma_n + 0.8333$
HL_2	$2.7533\sigma_n - 1.3$	$2.7067\sigma_n - 8.2667$	$2.8867\sigma_n + 0.8333$
HH_2	$8.8267\sigma_n - 26.9333$	$2.7067\sigma_n - 8.2667$	$2.8867\sigma_n + 0.8333$

TABLE IV: Optimized parameter values for the membership functions of the vector based colour extension.

parameter	value
thr_1	$1.5\sigma_n - 2.5$
T_1	$0.1667\sigma_n + 0.8333$
T_2	$0.6667\sigma_n + 11.6667$
t_1	0
t_2	$1.7\sigma_n + 2.5$

In our experiments, we have processed 6 different greyscale sequences ("Salesman", "Tennis", "Deadline", "Trevor", "Flower garden" and "Miss America") and 3 different colour sequences ("Salesman", "Chair" and "Tennis") with added Gaussian noise ($\sigma_n = 5, 10, 15, 20$). As a measure of objective dissimilarity between a filtered frame and the original one, the PSNR is used. This PSNR value is defined as:

$$MSE(I_0, I_f) = \frac{\sum_{c=1}^{C}\sum_{i=1}^{m}\sum_{j=1}^{n}(I_o(i,j,c) - I_f(i,j,c))^2}{n \cdot m \cdot C} ,$$

$$PSNR(I_0, I_f) = 10 \cdot \log_{10} \frac{S^2}{MSE(I_0, I_f)} ,$$

where I_o and I_f respectively denote the original and the filtered frame, each containing m rows and n columns of pixels and C channels ($C = 1$ for greyscale images and $C = 3$ for

colour images in the RGB colour space). S denotes the maximum possible greyscale value of a pixel (here $S = 255$).

For the colour sequences, we have also used a second measure, namely the normalized colour difference (NCD). The NCD is defined as:

$$NCD(I_o, I_f) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \|\Delta E_{LAB}\|}{\sum_{i=1}^{m} \sum_{j=1}^{n} \|E^*_{LAB}\|},$$

where I_o and I_f again stand for the original and the filtered frame respectively, each containing m rows and n columns of pixels,

$$\|\Delta E_{LAB}\| = \left((I_{o,L^*} - I_{f,L^*})^2 + (I_{o,a^*} - I_{f,a^*})^2 + (I_{o,b^*} - I_{f,b^*})^2\right)^{\frac{1}{2}}$$

and

$$\|E^*_{LAB}\| = \left((I_{o,L^*})^2 + (I_{o,a^*})^2 + (I_{o,b^*})^2\right)^{\frac{1}{2}},$$

where I_{o,L^*},I_{f,L^*},I_{o,a^*},I_{f,a^*},I_{o,b^*} and I_{f,b^*} respectively denote the L^*-component, the a^*-component and the b^*-component of the $L^*a^*b^*$-transform of the original and the filtered frame.

In Subsection VI A we compare our method with other state-of-the-art methods both in the pixel domain and the wavelet domain. Additionally, in Subsection VI B, we also test the use of our method for colour sequences. Subsection VI C, finally, tests the use of different fuzzy aggregators.

A. Comparison to Other State-Of-The-Art Methods

In this subsection, we compare our method to other state-of-the-art methods. We first compare our pixel domain method to other pixel domain methods and then do the comparison for the wavelet domain method.

1. Pixel Domain

The non-recursive (FMDAF) and recursive (RFMDAF) scheme of our fuzzy motion and detail adaptive filter in the pixel domain have been compared to the following well-known filters that also operate in the pixel domain (all with parameter values as suggested in the respective papers):

- the rational filter (Rational) [8],
- the 3D-KNN filter (KNN) [3] as an extension of the 2D-KNN filter [1, 2],
- the threshold averaging filter (THR) [3, 4],
- the motion and detail adaptive KNN filter (MDA-KNN) [3, 5],
- the recursive scheme of the multiple class averaging filter (RMCA) [7] (which performs better than the non-recursive one as shown in [7]).

Fig. 9 and Fig. 10 give the PSNR results for six test sequences processed with the above mentioned methods and for the noise levels $\sigma_n = 10$ and $\sigma_n = 15$ respectively. It can be seen that in terms of PSNR the FMDAF and RFMDAF filters outperform the other pixel domain methods. The MDA-KNN filter gives comparable results on the "Salesman" and "Deadline" sequences. Further, we also note that comparable results are found on the "Flower garden" sequence for the RMCA and the THR filters. For a visual comparison, the original "Trevor" sequence, the sequence with added Gaussian noise ($\sigma_n = 10$), and the noisy sequence processed by the different filters can be found on `http://www.fuzzy.ugent.be/tmelange/results/greyscale/pixel`. From the tests we also found that our method adapts better to motion than the RMCA method. In Fig.11 a part of the 18th frame of the "Trevor" sequence with added Gaussian noise ($\sigma_n = 10$) processed by the FMDAF method and the RMCA method is given. One clearly sees that our method has given a lower weight to those pixels from the previous frame situated in the fast moving arm.

Finally, we observed that the recursive scheme (RFMDAF) of the proposed filter removes slightly more noise than the non-recursive scheme (FMDAF), but this at the expense of little loss of spatial texture. Fig. 12 shows the 18th frame of the "Tennis" sequence with added Gaussian noise ($\sigma_n = 20$), processed by the FMDAF and by the RFMDAF. The texture on the wall is best preserved by the FMDAF method. But on the other hand, by looking carefully at the table, one sees that more noise is removed by the RFMDAF than by the FMDAF.

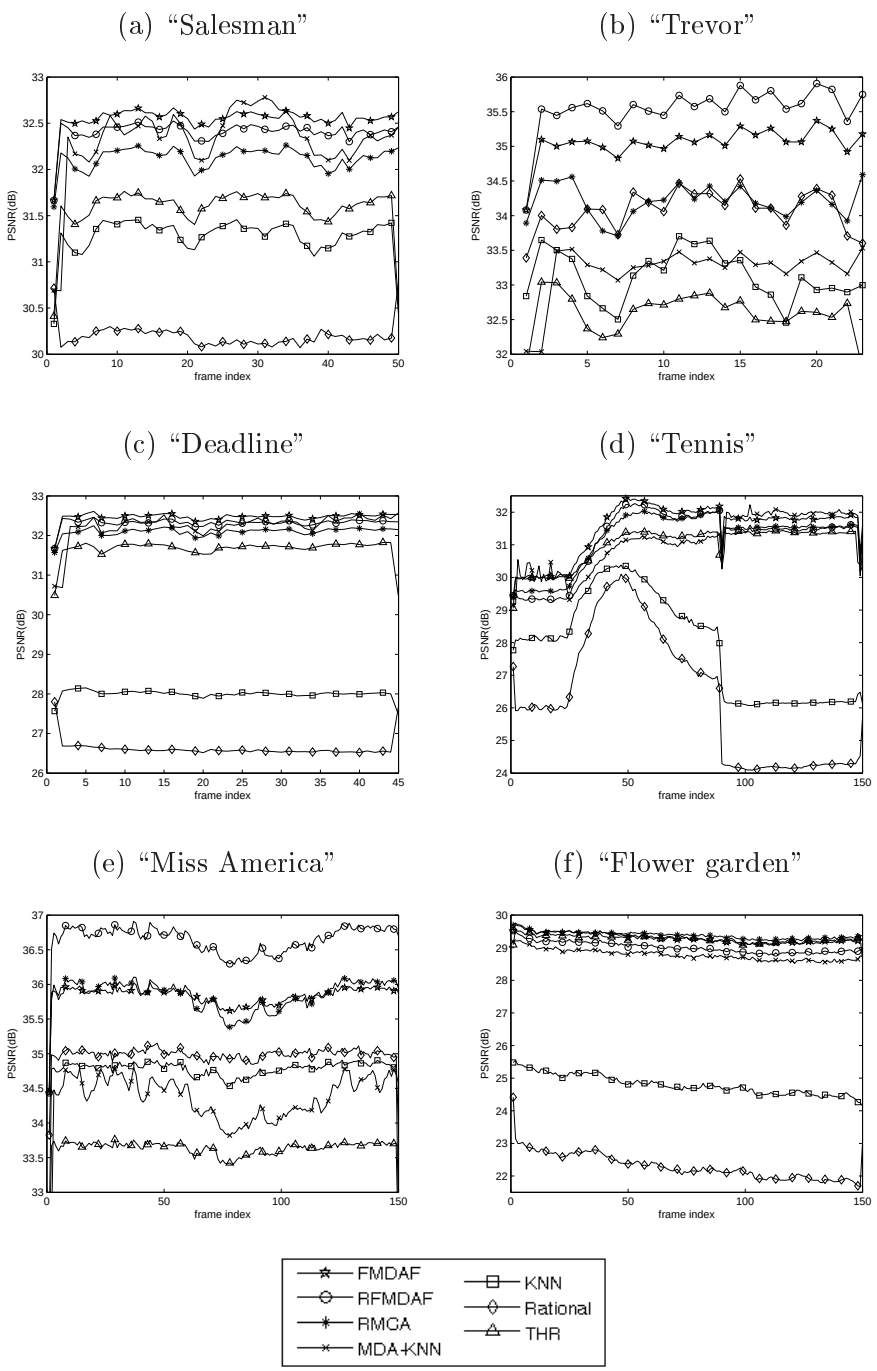

FIG. 9: Performance comparison for the pixel domain methods applied to different sequences with added Gaussian noise, $\sigma_n = 10$.

2. Wavelet Domain

The recursive (WRFMDAF) scheme of our wavelet domain method (which outperforms the non-recursive one) has been compared to the following methods (all with parameter

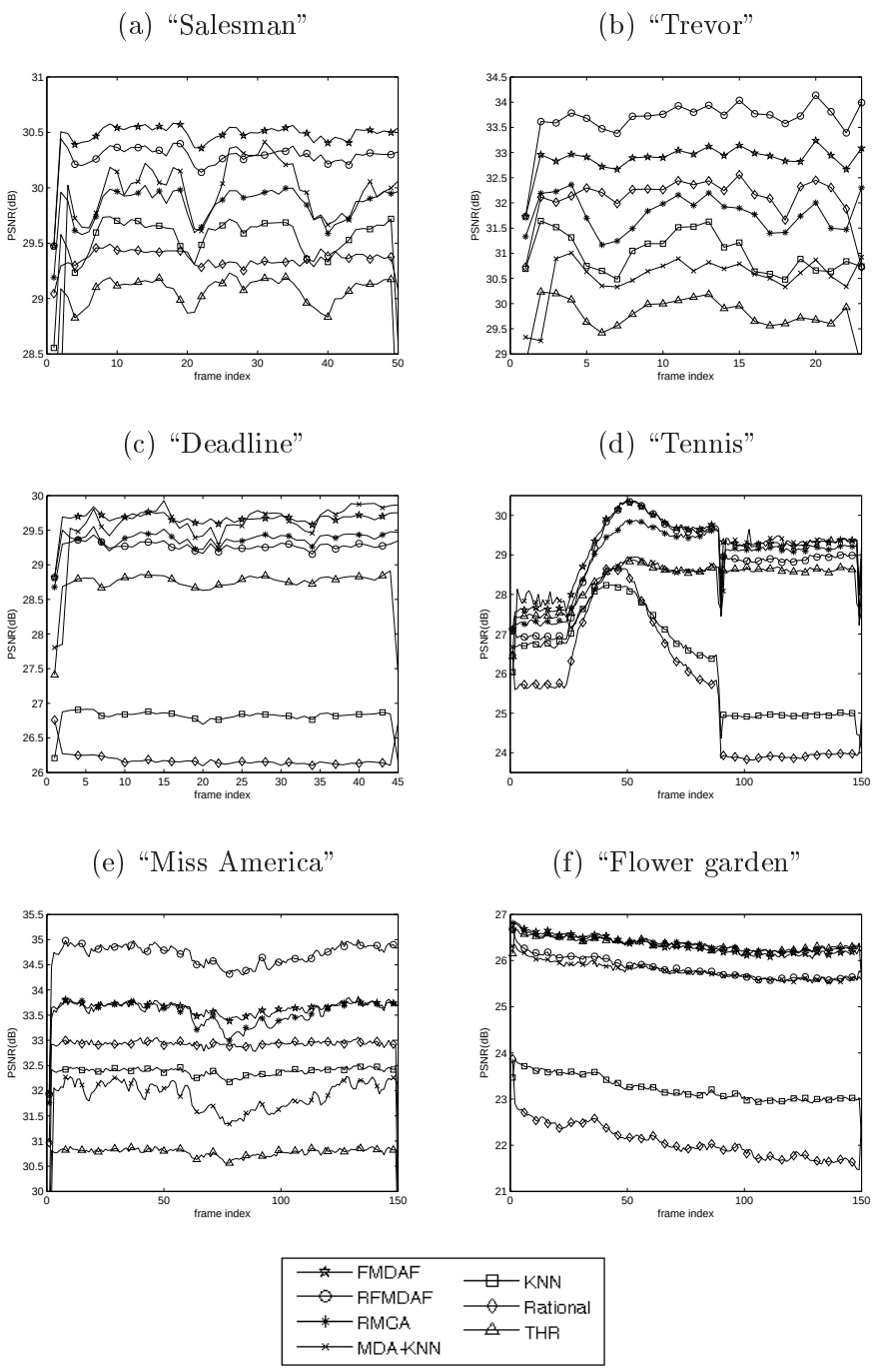

FIG. 10: Performance comparison for the pixel domain methods applied to different sequences with added Gaussian noise, $\sigma_n = 15$.

values as suggested in the respective papers):

- the recursive scheme of the wavelet domain multiple class averaging filter (WRMCA) [7] (non-decimated transform with the quadratic spline wavelet),

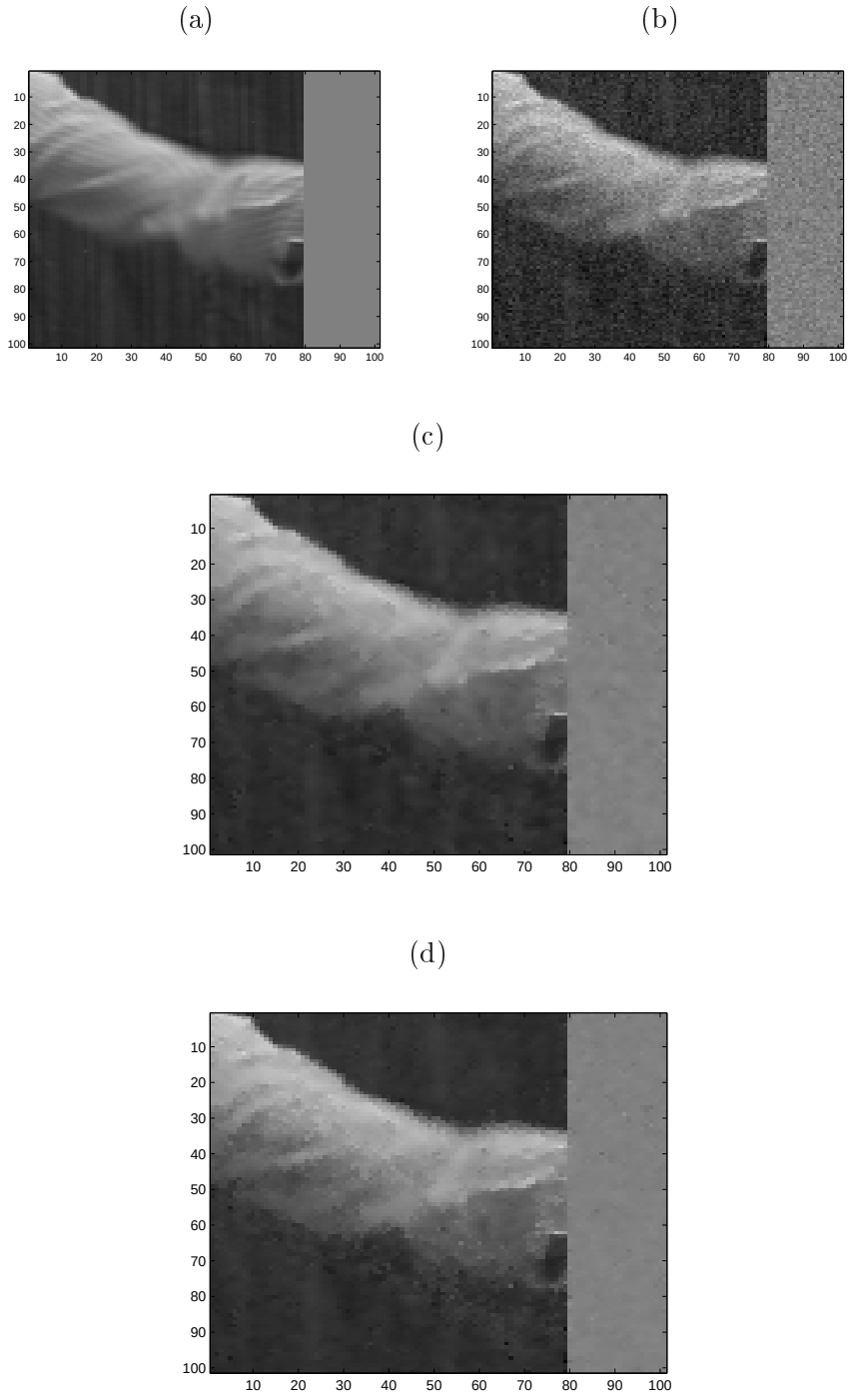

FIG. 11: Part of the 18th frame of the "Trevor" sequence (a) original; (b) with added Gaussian noise ($\sigma_n = 10$); (c) processed by the FMDAF method and (d) processed by the RMCA.

- the 3D wavelet transform filter (3DWF) [14] with the signal adaptive threshold from [11] (3-D dual-tree complex wavelet transform),

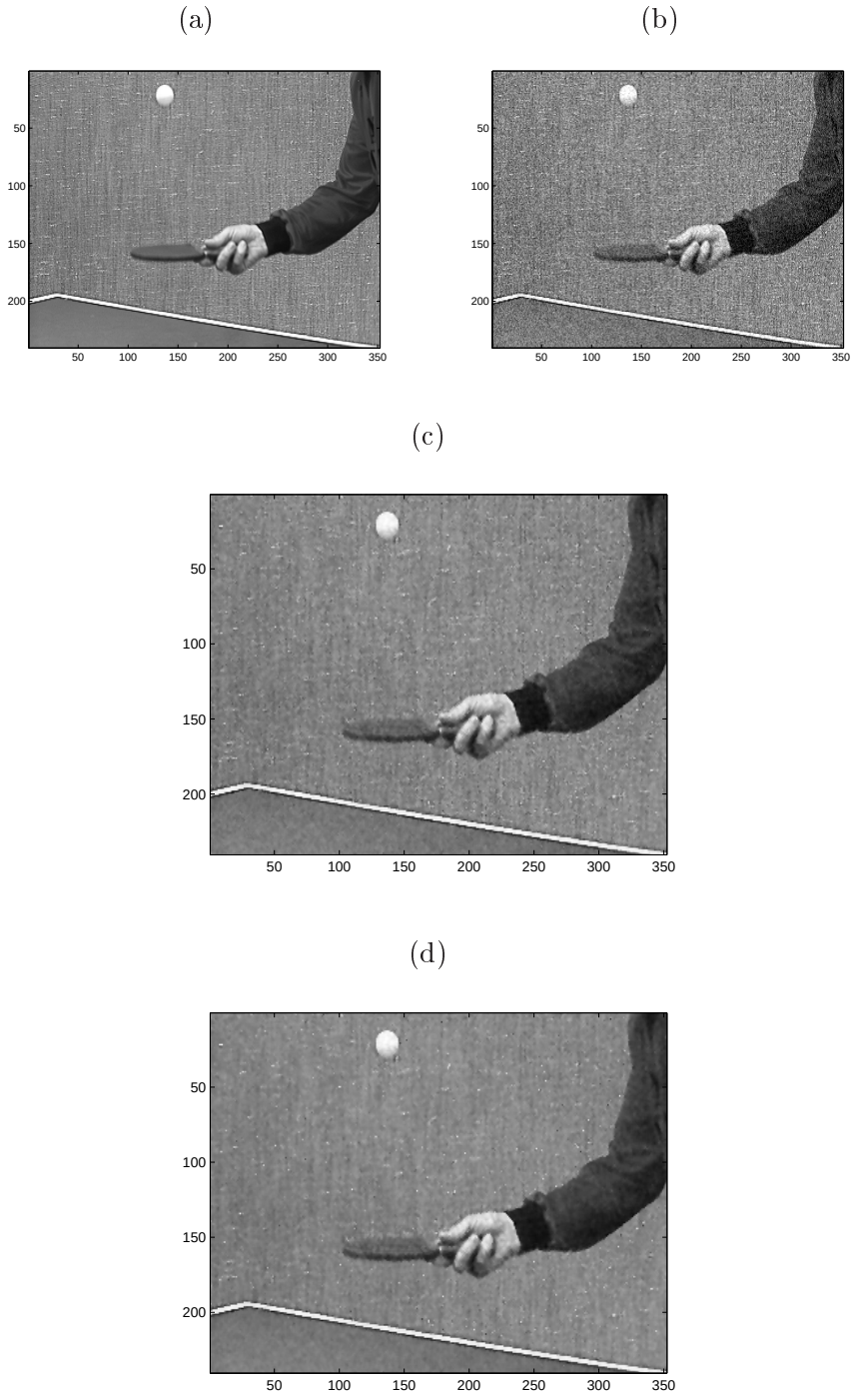

FIG. 12: 18th frame of the "Tennis" sequence (a) original; (b) with added Gaussian noise ($\sigma_n = 20$); (c) processed by the FMDAF method and (d) processed by the RFMDAF method.

- the sequential wavelet domain and temporal filter (SEQWT) [15] (non-decimated transform with the symmlet-8 wavelet),

- the adaptive spatio-temporal filter (ASTF) [16] (64-tap Johnston filter [37]),
- the video filter based on inter-frame statistical modelling of the wavelet coefficients (FISMW) [17] (decimated transform with the orthogonal symmlet-8 wavelet),
- the sparse 3D transform-domain collaborative filter for video (VBM3D) [26] (the decimated biorthogonal wavelet bior1.5 for the 2D-transform of the blocks and the decimated Haar-wavelet for the third dimension in the first step and the dct-transform (2D) and the decimated Haar-wavelet (third dimension) in the second step).

Fig. 13 and 14 gives the PSNR results for the processed "Salesman", "Trevor", "Deadline", "Tennis", "Miss America" and "Flower Garden" sequences. It can be seen that our method works best for a still camera filming possibly moving objects ("Salesman", "Trevor", "Deadline", "Miss America"). On such sequences our proposed wavelet based recursive WRFMDAF method clearly outperforms the ASTF method. We also see a better performance for the WRFMDAF than for the RMCA filter and similar results to those of the SEQWT filter. Taking into account that the degradations that result from using a decimated transform instead of a non-decimated one can reach up to 1 dB [15, 38], we might also conclude a similar performance for the FISMW filter. Still, more sophisticated filters like the VBM3D filter, consisting of two steps in which blocks are grouped by spatio-temporal predictive block-matching and each 3D group is filtered by a 3D transform domain shrinkage, and the complex 3D wavelet transform method 3DWF show better results in terms of PSNR than our proposed filter. For the "Flower garden" sequence, the received results are worse, because the performance of the additional time-recursive filtering in pixels where no motion is detected, will be reduced for a moving camera.

For a visual comparison, the original "Deadline" sequence, the sequence with added Gaussian noise ($\sigma_n = 10$), and the noisy sequence processed by the different filters can be found on `http://www.fuzzy.ugent.be/tmelange/results/greyscale/wavelet`. We see that a little less noise is removed by the RWFMDAF and WRMCA filters than by the SEQWT and FISMW filters, but on the other hand details are well preserved and less artefacts around the edges are introduced by the RWFMDAF filter.

It can be concluded that, for sequences obtained by a still camera, our method has a better performance in terms of PSNR than the other multiresolution filters of a similar complexity, but it is outperformed by some more sophisticated methods.

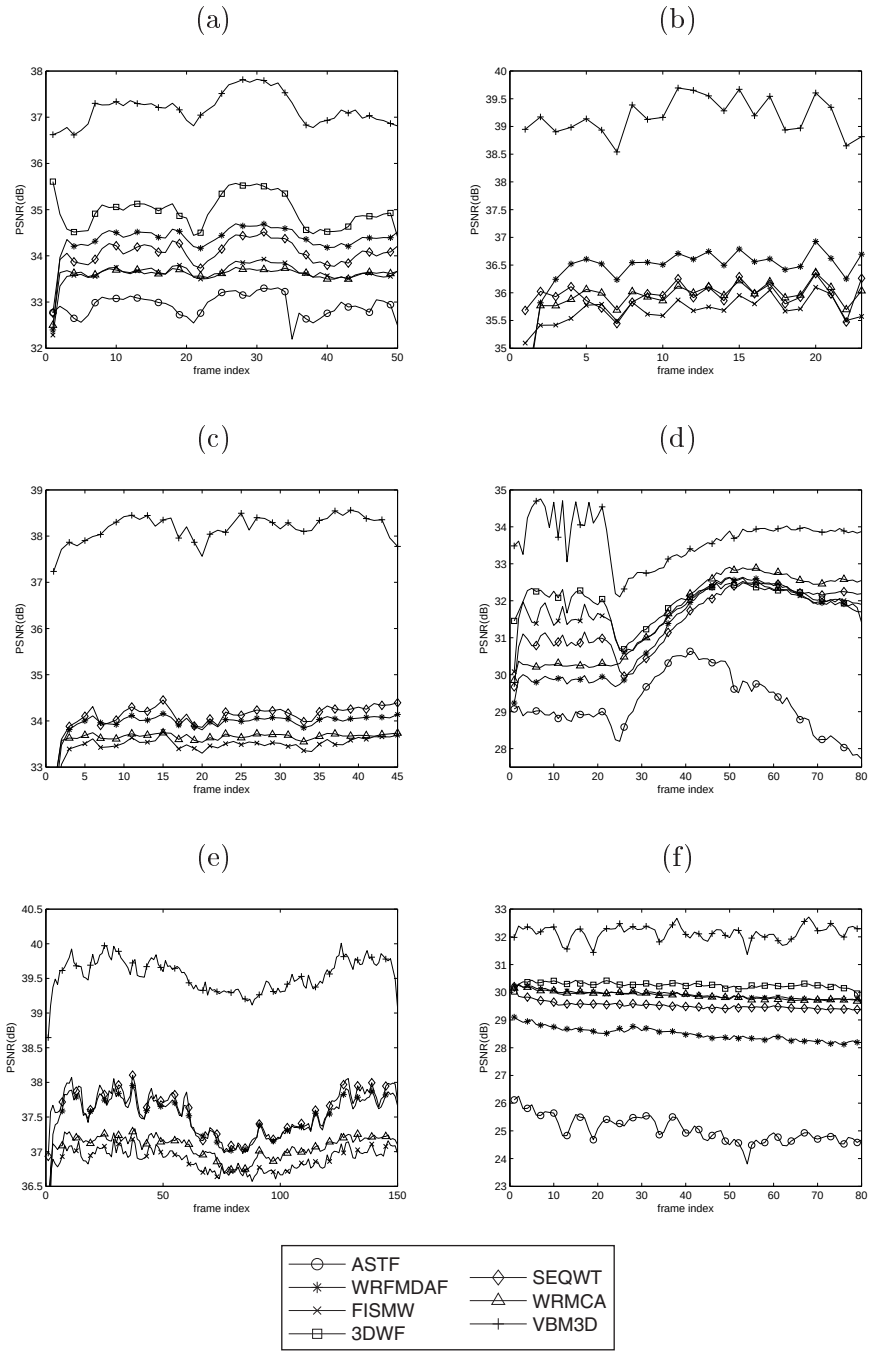

FIG. 13: Performance comparison for the wavelet domain methods applied to different sequences with added Gaussian noise ($\sigma_n = 10$): (a) "Salesman", (b) "Trevor", (c) "Deadline", (d) "Tennis", (e) "Miss America" and (f)"Flower Garden".

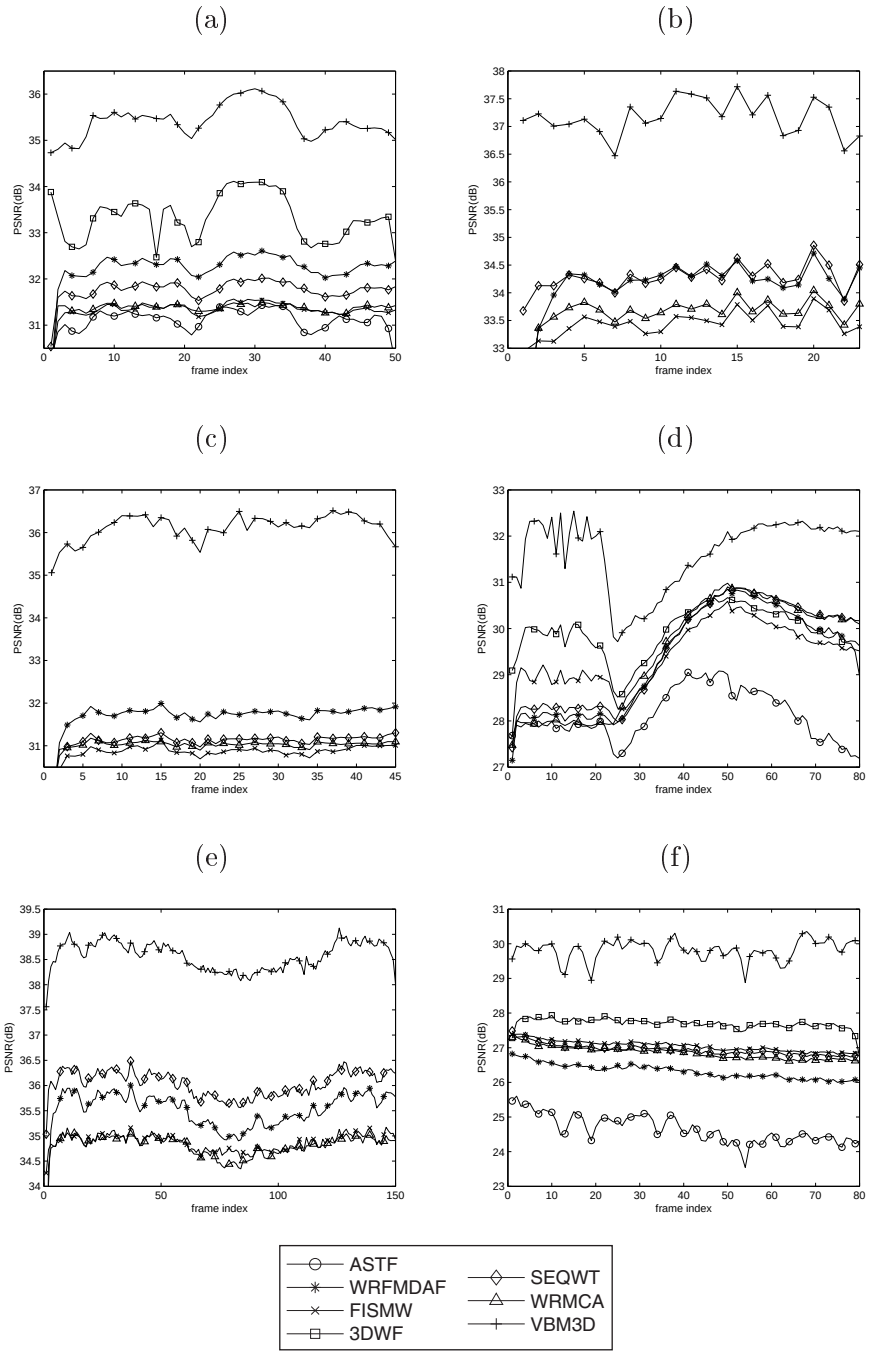

FIG. 14: Performance comparison for the wavelet domain methods applied to different sequences with added Gaussian noise ($\sigma_n = 15$): (a) "Salesman", (b) "Trevor", (c) "Deadline", (d) "Tennis", (e) "Miss America" and (f)"Flower Garden".

B. Processing of Colour Sequences

In this subsection, we test the use of our proposed method for colour sequences with equal noise levels on each colour band. We have compared our new vector based extension (FMDAF-$L^*a^*b^*$) from Section IV to the filtering scheme where the Y-component is filtered by the WRFMDAF method and where an additional spatial averaging is applied on the chrominance components U and V with an 3×3 filtering window (WRFMDAF-YUV). The results of this comparison are given in Tables V and VI. From Table V we see that the WRFMDAF-YUV method yields the best results in terms of PSNR. However, in terms of the NCD, which corresponds with human observation, it can be seen from Table VI that the best results are then obtained by the vector based FMDAF-$L^*a^*b^*$ method.

For a visual comparison, the original "Salesman" sequence, the sequence with added Gaussian noise ($\sigma_n = 10$) and the noisy sequence processed by respectively the WRFMDAF-YUV and FMDAF-$L^*a^*b^*$ method can be found on `http://www.fuzzy.ugent.be/tmelange/results/colour`. We see that a little more noise is removed by the wavelet domain WRFMDAF-YUV method, but also that more colour artefacts are introduced than by the FMDAF-$L^*a^*b^*$ method. This can for example be seen by looking carefully to the side of the phone.

We further also note that the FMDAF-$L^*a^*b^*$ method performs a little less good for the lowest noise level ($\sigma_n = 5$) in comparison to the performance for the other noise levels. The reason is that because of the two subfilters used in this method, there is a little too much averaging for this low noise level, resulting in a little more detail loss. For the other noise levels however, we see a more favorable compromise between noise removal and detail preservation.

C. The Use of Other Fuzzy Aggregators

In this subsection, we compare the performance of the proposed method, implemented with different triangular norms and conorms. In Table VII the results in terms of PSNR are given for different sequences processed with the RWFMDAF filter implemented with the suggested product norm and probabilistic sum conorm and other popular triangular norms and conorms. It can be seen that the performance of all aggregators are very comparable.

TABLE V: Comparison of the proposed colour extensions in terms of PSNR.

Sequence	**noise**	$PSNR_{av}$		
	level	*Input*	FMDAF-$L^*a^*b^*$	WRFMDAF-YUV
"Salesman"	$\sigma_n = 5$	34.16	35.20	**37.37**
	$\sigma_n = 10$	28.22	33.00	**33.64**
	$\sigma_n = 15$	24.82	30.73	**31.20**
	$\sigma_n = 20$	22.46	29.00	**29.35**
	$\sigma_n = 25$	20.66	27.71	**27.88**
"Chair"	$\sigma_n = 5$	34.17	37.35	**39.92**
	$\sigma_n = 10$	28.19	35.30	**35.67**
	$\sigma_n = 15$	24.71	32.77	**32.99**
	$\sigma_n = 20$	22.26	30.58	**30.89**
	$\sigma_n = 25$	20.39	**29.28**	29.16
"Tennis"	$\sigma_n = 5$	34.24	30.40	**33.40**
	$\sigma_n = 10$	28.26	29.20	**29.83**
	$\sigma_n = 15$	24.78	27.83	**27.92**
	$\sigma_n = 20$	22.31	26.60	**26.68**
	$\sigma_n = 25$	20.41	25.32	**25.59**

Only the weak norm and strong conorm seem to perform less good on some of the sequences. Therefore, we have chosen for the simple intermediate algebraic product and probabilistic sum.

VII. CONCLUSION

In this paper we have presented a new fuzzy motion and detail adaptive video filter intended for the reduction of additive white Gaussian noise in digital image sequences. The proposed algorithm has first been defined on greyscale images and in the pixel domain. In a next step we have adapted the algorithm to the wavelet domain. Finally, we have also extended the method to handle colour image sequences.

TABLE VI: Comparison of the proposed colour extensions in terms of NCD.

Sequence	**noise**	NCD_{av}		
	level	*Input*	FMDAF-$L^*a^*b^*$	WRFMDAF-YUV
"Salesman"	$\sigma_n = 5$	0.1041	**0.0524**	0.0531
	$\sigma_n = 10$	0.2077	**0.0657**	0.0829
	$\sigma_n = 15$	0.3050	**0.0813**	0.1121
	$\sigma_n = 20$	0.3929	**0.0959**	0.1397
	$\sigma_n = 25$	0.4724	**0.1097**	0.1652
"Chair"	$\sigma_n = 5$	0.0300	0.0152	**0.0122**
	$\sigma_n = 10$	0.0599	**0.0183**	0.0215
	$\sigma_n = 15$	0.0899	**0.0221**	0.0309
	$\sigma_n = 20$	0.1198	**0.0263**	0.0404
	$\sigma_n = 25$	0.1497	**0.0301**	0.0498
"Tennis"	$\sigma_n = 5$	0.0434	0.0446	**0.0339**
	$\sigma_n = 10$	0.0857	0.0490	**0.0475**
	$\sigma_n = 15$	0.1270	**0.0545**	0.0601
	$\sigma_n = 20$	0.1677	**0.0605**	0.0724
	$\sigma_n = 25$	0.2079	**0.0672**	0.0848

Experimental results show that our pixel domain greyscale method and the wavelet domain extension outperform respectively other state-of-the-art pixel domain filters and other state-of-the-art wavelet domain filters of a comparable complexity in terms of PSNR. For the processing of colour images we conclude that the proposed FMDAF-$L^*a^*b^*$ colour extension is a good alternative for the filtering scheme in the YUV colour space.

As future work we will include colour information into the fuzzy rules directly instead of working with colour vectors and we will try to find a framework for the denoising of video sequences corrupted with other types of noise such as impulse noise and α-stable noise.

Acknowledgement. This research was financially supported by the FWO project

TABLE VII: Comparison of the different aggregators.

Sequence ($\sigma_n = 10$)	$PSNR_{av}$			
	algebraic product/ probabilistic sum	minimum/ maximum	weak/ strong	Łukasiewicz
"Salesman"	**34.37**	34.36	34.10	34.36
"Trevor"	36.41	**36.42**	35.55	36.29
"Deadline"	33.95	33.91	33.74	**33.98**
"Tennis"	31.44	31.37	31.47	**31.55**
"Miss America"	**37.48**	**37.48**	36.76	37.42
"Flower Garden"	28.27	28.13	**28.50**	28.44

G.0667.06 of Ghent University. A. Pizurica is a postdoctoral research fellow of FWO, Flanders. The authors would like to thank Prof. Selesnick from the Polytechnic University, New York, for providing them with the processed video sequences for the 3DWF algorithm, which have been used for the comparison. They would also like to give a special thanks to Dr. A.M. Tourapis for providing them with the processed sequences by the ASTF algorithm. Finally, a special thanks goes to S.M. Mahbubur Rahman from Concordia University, Montréal, for providing the code of the FISMW filter, which has been used for the comparison.

[1] Davis, L., Rosenfeld, A.: "Noise cleaning by iterated local averaging", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 8, pp. 705-710, 1978.

[2] Mitchell, H., Mashkit, N.: "Noise smoothing by a fast k-nearest neighbor algorithm", *Signal Processing: Image Communication*, Vol. 4, pp. 227-232, 1992.

[3] Zlokolica, V.: *Advanced nonlinear methods for video denoising*, PhD thesis, Chapter 3, Section 3, Ghent University, Ghent, Belgium, 2006.

[4] Lee, J.S.: "Digital image smoothing and the sigma filter" *Computer Vision, Graphics, and Image Processing.*, Vol. 24, pp. 255-269. Nov. 1983

[5] Zlokolica, V., Philips, W.: "Motion-detail adaptive k-nn filter video denoising", Report 2002, http://telin.ugent.be/∼vzlokoli/Report2002vz.pdf.

[6] Zlokolica, V., Pizurica, A., Philips, W.: "Video denoising using multiple class averaging with multiresolution", *International Workshop VLBV03, Lecture Notes in Computer Science of Springer Verlag*, Vol. 2849, pp. 172-179, Madrid, Spain, 2003.

[7] Zlokolica, V.: *Advanced nonlinear methods for video denoising*, PhD thesis, Chapter 5, Ghent University, Ghent, Belgium, 2006.

[8] Cocchia, F., Carrato, S., Ramponi, G.: "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.

[9] Yin, H.B., Fang, X.Z., Wei, Z., Yang, X.K.: "An improved motion-compensated 3-D LLMMSE filter with spatio-temporal adaptive filtering support", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, no. 12, pp. 1714-1727, Dec. 2007.

[10] Guo, L., Au, O.C., Ma, M., Liang, Z.: "Temporal video denoising based on multihypothesis motion compensation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, no. 10, pp. 1423-1429, Oct. 2007.

[11] Sendur, L., Selesnick, I.W.: "Bivariate shrinkage functions for wavelet based denoising exploiting interscale dependency", *IEEE Transactions on Image Processing*, Vol. 50, no. 11, pp. 2744-2756, Nov. 2002.

[12] Balster, E.J., Zheng, Y.F., Ewing, R.L.: "Feature-based wavelet shrinkage algorithm for image denoising", *IEEE Transactions on Image Processing*, Vol. 14, no. 3, pp. 2024-2039, 2005.

[13] Rajpoot, N., Yao, Z., Wilson, R.: "Adaptive wavelet restoration of noisy video sequences", *Proc. International Conference on Image Processing, ICIP*, pp. 957-960, Singapore, 2004.

[14] Selesnick, I.W., Li, K.Y.: "Video denoising using 2d and 3d dual-tree complex wavelet transforms", *Proc. SPIE Wavelet Applicat. Signal Image Process.*, San Diego, CA, pp. 607-618, Aug. 2003.

[15] Pizurica, A., Zlokolica, V., Philips, W.:"Noise reduction in video sequences using wavelet-domain and temporal filtering", *Proc. SPIE Conf. Wavelet Applicat. Industrial Process.*, Providence, RI, pp. 48-59, Oct. 2003.

[16] Cheong, H., Tourapis, A., Llach, J., Boyce, J.: "Adaptive spatio-temporal filtering for video de-noising", *IEEE International Conference on Image Processing*, pp. 965-968, Singapore,

2004.

[17] Mahbubur Rahman, S.M., Omair Ahmad M., Swamy, M.N.S.:"Video denosing based on inter-frame statistical modeling of wavelet coefficients", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, no. 2, pp. 187-198, Feb. 2007.

[18] Balster, E.J., Zheng, Y.F., Ewing, R.L.: "Combined spatial and temporal domain wavelet shrinkage algorithm for video denoising", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, no. 2, pp. 220-230, Feb. 2006.

[19] Zlokolica, V., Pizurica, A., Philips, W.: "Wavelet-domain video denoising based on reliability measures", *IEEE Transactions on circuits and systems for video technology*, Vol.16, no. 8, pp. 993-1007, Aug. 2006.

[20] Jovanov, L., Pizurica, A., Zlokolica, V., Schulte, S., Kerre, E.E., Philips, W.: "Combined wavelet domain and temporal filtering complient with video codec", *IEEE Internat. Conf. on Acoust., Speech and Signal Process.*, ICASSP'07 , Honolulu, Hawaii, USA, Apr. 2007.

[21] Jin, F., Fieguth, P., Winger, L.: "Wavelet video denoising with regularized multiresolution motion estimation" *EURASIP Journal on Applied Signal Processing*, Vol 2006, Issue 1, Jan. 2006.

[22] Lian, N.-X., Zagorodnov, V., Tan, Y.-P.: "Video denoising using vector estimation of wavelet coefficients", *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS 2006*, May 2006.

[23] Chalidabhongse, J., Jay Kuo, C.-C.: "Fast motion vector estimation using multiresolution-spatio-temporal correlations", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol 16, no. 8, pp. 993-1007, 1997.

[24] Tekalp, M.: *Digital Video Processing.* Prentice Hall, Inc., 1995.

[25] Wiegand, T., Sullivan, G., Bjøntegaard, G., Luthra, A.: "Overview of the h.264/avc video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol 13, no. 7, pp. 560-576, 2003.

[26] Dabov, K., Foi, A., Egiazarian, K.: "Video denoising by sparse 3-D transform-domain collaborative filtering", *Proc. 15th European Signal Processing Conference, EUSIPCO 2007*, Poznan, Poland, Sept 2007.

[27] Russo, F.: "Hybrid neuro-fuzzy filter for impulse noise removal", *Pattern Recognition*, Vol. 32, pp. 1843-1855, 1999.

[28] Yildirim, M.T., Yüksel, M.E.: "A type-2 fuzzy logic filter for detail-preserving restoration of digital images corrupted by impulse noise", *Lecture notes in computer science*, Vol. 4678, pp. 485-496, 2007.

[29] Schulte, S., De Witte, V., Kerre, E.E.: "A fuzzy noise reduction method for color images", *IEEE Transactions on Image Processing*, Vol 16, no. 5, May 2007.

[30] Bellers, E.B., De Haan, G.: *De-interlacing: A Key Technology for Scan Rate Conversion*, Elsevier Science B.V., Sara Burgerhartstraat, Amsterdam, 2000.

[31] Zadeh, L.A.: "Fuzzy Sets", *Information and Control*, Vol 8, No. 5, pp. 338-353, 1965.

[32] Donoho, D., Johnstone, I.: "Ideal spatial adaptation by wavelet shrinkage", *Biometrika*, Vol. 8, pp. 425-455, 1994.

[33] Weber, S.: "A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms", Fuzzy Sets and Systems, Vol. 11, no. 2, pp. 115-134, 1983.

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

[35] Slater, J.: *Modern Television Systems to HDTV and beyond*, Pitman, London, 1991.

[36] Sangwine, S.J. and Horne, R.E.N.: *The colour image processing handbook*, Chapman&Hall, London, 1998.

[37] Johnston, J.D.: "A filter family designed for use in quadrature mirror filter banks", *Proceedings ICASSP*, Vol. 1, pp. 291-294, Denver, CO, Apr. 1980.

[38] Chang, S.G., Yu, B., Vetterli, M.:"Spatially adaptive wavelet thresholding with context modelling for image denoising", *IEEE Transactions on Image Processing*, Vol. 9, pp. 1522-1531, Sept. 2000.

Tom Mélange was born in Kortrijk, Belgium, in 1984. He received the M.Sc. degree in Mathematics from Ghent University, Ghent, Belgium, in 2006. In October 2006, he joined the Department of Applied Mathematics and Computer Science, Ghent University, where he is a member of the Fuzziness and Uncertainty Modelling Research Unit working towards

the Ph.D. degree with a thesis on fuzzy techniques for noise reduction in video under the promotorship of Prof. E. E. Kerre.

Stefan Schulte was born in Dortmund, Germany, in 1979. After secondary school he studied Computer Science at the University of Ghent. In 2003 he obtained a M.Sc. degree. In September 2003, he joined the Department of Applied Mathematics and Computer Science, Ghent University, where he was a member of the Fuzziness and Uncertainty Modelling Research Unit. He obtained the Ph.D. 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. Currently he's working as an R&D engineer at Traficon NV, a reference for traffic video detection.

Valérie De Witte was born in Ghent, Belgium, in 1981. She received the M.Sc. degree in Mathematics from Ghent University, Ghent, Belgium, in 2003. In September 2003, she joined the Department of Applied Mathematics and Computer Science, Ghent University, where she was a member of the Fuzziness and Uncertainty Modelling Research Unit. She obtained the Ph.D. degree in 2007 with a thesis on colour mathematical morphology under the promotorship of Prof. E. E. Kerre. Currently she's working as a researcher at VRT, Flanders' public service broadcaster.

Mike Nachtegael was born on February 16, 1976. After secondary school he went to Ghent University, where he obtained a M.Sc. in Mathematics in 1998. In the same year he joined the Department of Applied Mathematics and Computer Science, where he is a member of the Fuzziness and Uncertainty Modelling Research Unit. In May 2002 he obtained a Ph.D. in Mathematics, on the topic of Fuzzy Techniques in Image Processing. Currently, he has the position of doctor-assistant in his Department.

Etienne Kerre was born in Zele, Belgium on May 8, 1945. He obtained his M.Sc. degree in Mathematics in 1967 and his Ph.D. in Mathematics in 1970 from Ghent University. Since 1984, he has been a lector, and since 1991, a full professor at Ghent University. He is a referee for more than 30 international scientific journals, and also a member of the editorial board of international journals and conferences on fuzzy set theory. He was an honorary chairman at various international conferences. In 1976, he founded the Fuzziness and Uncertainty Modelling Research Unit (FUM) and since then his research has been focused on the modelling 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 of him. Over the years he has also been a promotor of 20 Ph.Ds on fuzzy set theory. His current research interests include fuzzy and intuitionistic fuzzy relations, fuzzy topology, and fuzzy image processing. He has authored or co-authored 11 books, and more than 300 papers appeared in international refereed journals and proceedings.