A Vector Based Fuzzy Filter for Colour Image Sequences

Tom Mélange, Mike Nachtegael and **Etienne E. Kerre**
Department of Applied Mathematics and Computer Science,
Fuzziness and Uncertainty Modelling Research Unit,
Krijgslaan 281 (Building S9), 9000 Gent, Belgium
Tom.Melange@UGent.be
`http://www.fuzzy.ugent.be`

Abstract

This paper introduces a new vector based filter for the removal of additive gaussian noise in colour image sequences. The proposed method consists of two subfilters of which the first is a colour extension of our previously presented fuzzy logic based motion and detail adaptive filter (FMDAF) for greyscale image sequences. The additional second subfilter is a 3D adaptation of the colour restorating subfilter from a recent fuzzy noise reduction method for colour images. Experimental results show that in terms of average PSNR and NCD, the proposed colour extension outperforms the usual colour extension in which the Y component in the YUV transform is filtered with the original greyscale method. However, better objective measures need to be found for correspondence to visual observations.

Keywords: Video, colour, noise, Gaussian.

1 Introduction

In today's world image sequences can be found in almost all kind of areas. A few examples are e.g. broadcasting, video-phone, internet applications (chat, skype, YouTube, ...), traffic observations, surveillance systems, autonomous navigation and so on.

The used sequences are however often affected by noise due to bad acquisition, transmission, recording and/or compression. In many applications, the noise can be modelled by an additive white Gaussian noise model of zero mean and variance σ^2:

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

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 total number of pixels per frame. In this paper we concentrate on this noise model.

Two goals can be aimed at by noise filtering, namely: (i) a visual improvement and (ii) an improvement in the further analysis or coding of the sequences. To reach these purposes, it is important to find a good balance between noise removal and preservation of fine details.

In literature, several filtering methods for greyscale image sequences can be found, e.g. [1, 3, 4, 5, 6, 7, 9, 11]. A first straightforward way to extend a given greyscale method to colour image sequences modelled in the RGB colour space, is to filter each of the colour bands R, G and B separately. Applying this approach, colour artefacts are introduced because of the neglected correlation between the different colour bands. Therefore, usually a second approach is applied. In this approach only the luminance component Y of the YUV-transform is filtered with the given greyscale method, possibly with an additional averaging of the chrominance components U and V. In the first subfilter of the proposed framework, we opt for a vector

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 1191–1198
Torremolinos (Málaga), June 22–27, 2008

based approach in which each pixel is treated as a colour vector and none of its components are used separately. In this first subfilter our previous work (FMDAF filter) presented in [6] is extended to handle such colour vectors. The filter from [6] is inspired by the ideas behind the multiple class averaging filter [11], but uses them in a fuzzy logic framework. Fuzzy set theory and fuzzy logic replace binary decisions by gradual transitions, which are much more appropriate for handling complex systems and for the processing of human knowledge in the form of fuzzy if-then rules. The underlying ideas are: (i) use only pixels from the current frame in the case of motion detection to prevent spatio-temporal blur; (ii) filter less in the case that large spatial activity (image details) is detected. This means that more noise will be left in such areas. However, the human eye is less sensitive for the high spatial frequencies corresponding to large spatial activity [2]. When no image details are detected, a strong filtering can be performed to remove as much noise as possible.

Additionally to the first subfilter, a 3D extension of the colour restorating second subfilter from [8] is applied.

The experimental results show that the proposed colour video denoising framework performs very well in terms of average PSNR and NCD. It is however also discussed that these measures, which are still used on a wide scale as standard measures for image comparison, do not always correspond to visual observations and that better measures are needed.

The structure of the paper is as follows: The proposed filtering framework for the denoising of colour video is explained in Section 2. Additionally, in Section 3 the experimental results are presented. Finally, the paper is concluded in Section 4.

2 The Proposed Denoising Framework

In this section, the proposed filtering framework is explained. The framework consists of two subfilters which are defined in Subsection 2.1 and Subsection 2.2 respectively. In this paper we assume the frames of the image sequence to be modelled in the RGB colour space, as it is the case in most image and video processing applications. In this colour space, colours are the results of adding together the primary components red, green and blue in different proportions (each given by a value in $[0, 255]$). A colour can thus be represented by a 3D vector, where the first, second and third component respectively indicate the amount of red, green and blue in the given colour.

2.1 First Subfilter

The first subfilter makes use of 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 shown Fig. 1. This window is moved through the frame step by step from top left to bottom right. In each step the current central position in the window is filtered by averaging the noise over neighbouring pixels that are similar and thus probably belong to the same object. The central position in the window is denoted by $(\mathbf{r}, t)$, with $\mathbf{r} = (x, y)$ the spatial position of the pixel in the frame and t the temporal position of the frame in the sequence. Additionally, $(\mathbf{r'}, t')$, with $\mathbf{r'} = (x + k, y + l)$, $(-1 \leq k, l \leq 1)$ and $t' = t$ or $t' = t - 1$, denotes an arbitrary pixel position in the filtering window. We further

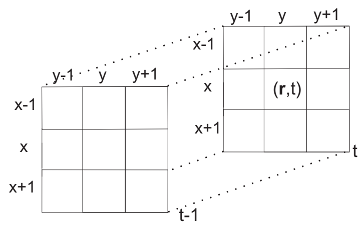

Figure 1: The $3 \times 3 \times 2$ filtering window consisting of 3×3 pixels in the current frame and 3×3 pixels in the previous frame.

denote the noisy input sequence and the output of the first fuzzy subfilter by I_n and I_f respectively.

The filtered value $I_f(\mathbf{r}, t)$ of the first subfilter for the central pixel in the window is defined as a weighted mean of the pixels in the $3 \times 3 \times 2$

window:

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

where the weights $W(\mathbf{r'},t',\mathbf{r},t)$ correspond to the activation degree of one of the Fuzzy Rules 2 and 3 given below in Subsection 2.1.2. These fuzzy rules are based on a detail value $d(\mathbf{r},t)$, a difference value $\Delta(\mathbf{r'},t',\mathbf{r},t)$ and a motion value $m(\mathbf{r},t)$. These values are vector extensions of the values used in [6] which we adopted from [11].

- The detail value $d(\mathbf{r},t)$ is equal to the standard deviation of the 3×3 pixels of the sliding window belonging to the current frame:

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

$$d(\mathbf{r},t) = \Big(\frac{1}{9} \sum_{\mathbf{r'}} \|I_n(\mathbf{r'},t) - I_{av}(\mathbf{r},t)\|_2^2\Big)^{\frac{1}{2}}.$$

- The difference value $\Delta(\mathbf{r'},t',\mathbf{r},t)$ in the fuzzy rules is defined as:

$$\Delta(\mathbf{r'},t',\mathbf{r},t) = \big\|I_n(\mathbf{r'},t') - I_n(\mathbf{r},t)\big\|_2.$$

- The motion value $m(\mathbf{r},t)$ used for the filtering is finally determined as:

$$m(\mathbf{r},t) = \|I_n(\mathbf{r},t) - I_n(\mathbf{r},t-1)\|_2.$$

2.1.1 Fuzzy Sets and Fuzzy Rules

To be able to express whether the above defined values are "large", we introduce fuzzy sets. A fuzzy set F in a universe X is characterized by a $X \mapsto [0,1]$ mapping μ_F, called the membership function. This mapping associates with every element $x \in X$ a membership degree $\mu_F(x)$ of x in F. For the above introduced values, we respectively introduce a fuzzy set "large detail value", "large difference" and "large motion value". A membership degree equal to one for e.g. $d(\mathbf{r},t)$ in the fuzzy set "large detail value" means that this detail value is large for sure. A membership degree equal to zero means that we are certain that the detail value is not large. Membership degrees between zero and one make it possible to have a gradual transition between those two cases and give an indication of whether the detail value is rather large or not. The membership functions of the three above fuzzy sets are respectively denoted by μ_d, μ_Δ and μ_m and are depicted in Fig. 2, 3 and 4. The parameters thr_1, thr_2, T_1, T_2, t_1 and t_2 that define the exact form of these functions are experimentally optimized in terms of PSNR, by letting them vary over an interval of values, for different sequences with different characteristics and for different noise levels. These results indicated linear relationships between the optimal parameter values and the noise level. The best fitting

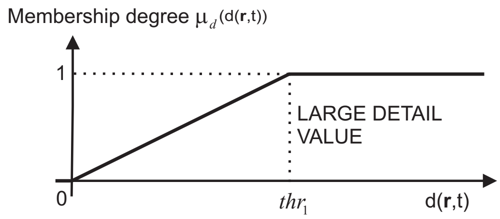

Figure 2: The membership function μ_d of the fuzzy set "large detail value".

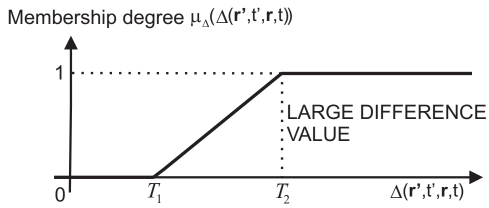

Figure 3: The membership function μ_Δ of the fuzzy set "large difference".

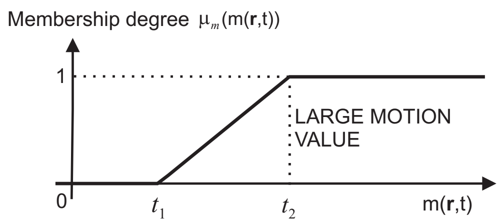

Figure 4: The membership function μ_m of the fuzzy set "large motion value".

lines through the observations can be found in Table 1. In this table, σ_n stands for the standard deviation of the Gaussian noise which is assumed equal in the three colour bands.

Table 1: Optimized parameter values for the membership functions.

parameter	optimal value
thr_1	$0.22\sigma_n - 1.8$
thr_2	$2.585\sigma_n - 4.875$
T_1	$1.03\sigma_n - 7.9$
T_2	$3.34\sigma_n + 3.65$
t_1	$0.12\sigma_n - 1.2$
t_2	$3.665\sigma_n - 2.225$

As mentioned earlier, the weights $W(\mathbf{r'}, t', \mathbf{r}, t)$ in (1) are defined as the activation degree of a fuzzy rule. Such a fuzzy rule has the general form "IF A THEN B". A is called the premise (or antecedent) and B the consequent. Both A and B are (collections of) propositions containing linguistic variables (e.g. "large"). Propositions can be connected by AND and OR operators or preceeded by NOT operators, corresponding to respectively the intersection and union of two fuzzy sets and the complement of a fuzzy set.

To determine the membership degree of an element y in the intersection of two fuzzy sets F_1 and F_2 in Y, a triangular norm T [10] is used in fuzzy logic. This norm T maps the membership degrees of the element y in the fuzzy sets F_1 and F_2 onto its membership degree in the fuzzy set $F_1 \cap F_2$: $\mu_{(F_1 \cap F_2)}(y) = T(\mu_{F_1}(y), \mu_{F_2}(y))$, $\forall y \in Y$.

To obtain the membership degree of an element y in the union of F_1 and F_2 from the membership degrees in F_1 and F_2 in fuzzy logic a triangular conorm S [10] is used: $\mu_{(F_1 \cup F_2)}(y) = S(\mu_{F_1}(y), \mu_{F_2}(y))$,$\forall y \in Y$.

Finally, the membership degree of an element y in the complement of a fuzzy set F in Y, given its membership degree in F, is determined using an involutive negator [10]: $\mu_{(co(F))}(y) = N(\mu_F(y))$, $\forall y \in Y$.

For the results in this paper, we have chosen for the algebraic product and the probabilistic sum as triangular norm and conorm respectively and for the well-known standard negator $N_s(x) = 1 - x$, $\forall x \in [0, 1]$. There is however no remarkable difference to the results obtained by using other norms (e.g. the minimum) and conorms (e.g. the maximum).

As an example, we consider the following fuzzy rule:

Fuzzy Rule 1 `IF` (u *is* `U AND` v *is* V) `OR` w *is* `NOT W THEN` z *is* `Z`.

The membership degree $\mu_Z(z)$ of z in Z, corresponding to the activation degree of the rule, is then calculated as:

$$\begin{aligned}\mu_Z(z) &= (\mu_U(u) \cdot \mu_V(v)) + (1 - \mu_W(w)) - \\ &\quad (\mu_U(u) \cdot \mu_V(v)) \cdot (1 - \mu_W(w)).\end{aligned}$$

2.1.2 Weight Determination

Depending on whether the window pixel at position $(\mathbf{r'}, t')$ lies in the current ($t' = t$) or in the previous ($t' = t - 1$) frame, the weight $W(\mathbf{r'}, t', \mathbf{r}, t)$ in (1) is determined as the activation degree of one of the fuzzy rules given below. The rules remain the same as in our previous work [6], but with adapted detail, difference and motion values and they are now used to assign weights to colour vectors instead of grey values.

Fuzzy Rule 2 *Determining the membership degree in the fuzzy set "large weight" of the weight $W(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ for the pixel at position $\boldsymbol{r'}$ in the current frame ($t' = t$) of the window with central pixel position $(\boldsymbol{r}, t)$:*

`IF` (*the detail value* $d(\boldsymbol{r}, t)$ *is* `LARGE AND`
$\Delta(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* `NOT LARGE`)
`OR` (*the detail value* $d(\boldsymbol{r}, t)$ *is* `NOT LARGE`)
`THEN` *the pixel at position* $\boldsymbol{r'}$ *has a* `LARGE` *weight* $W(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *in (1).*

Fuzzy Rule 3 *Determining the membership degree in the fuzzy set "large weight" of the weight $W(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ for the pixel at position $\boldsymbol{r'}$ in the previous frame ($t' = t - 1$) of the window with central pixel position $(\boldsymbol{r}, t)$:*

`IF` ((*the detail value* $d(\boldsymbol{r}, t)$ *is* `LARGE AND`

$\Delta(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE)

OR (*the detail value* $d(\boldsymbol{r}, t)$ *is* NOT LARGE))

AND *the motion value* $m(\boldsymbol{r}, t)$ *is* NOT LARGE

THEN *the pixel at position* $\boldsymbol{r'}$ *has a* LARGE *weight* $W(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *in (1).*

The weight $W(\mathbf{r'}, t', \mathbf{r}, t)$ (corresponding to the activation degree of one of the two rules) is thus equal to its membership degree in a fuzzy set "large weight", determined by the membership function depicted in Fig. 5. This membership degree or thus the weight

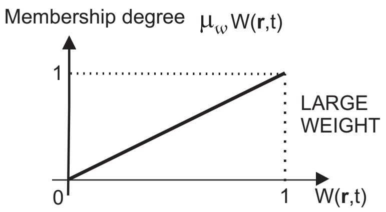

Figure 5: The membership function μ_w of the fuzzy set "large weight".

$W(\mathbf{r'}, t', \mathbf{r}, t)$ in (1) is more precisely given by

$$\omega \cdot (1-\theta) + (1-\omega) - \omega \cdot (1-\theta) \cdot (1-\omega),$$

for pixel positions in the window belonging to the current frame and by

$$\Big(\omega \cdot (1-\theta) + (1-\omega) - \omega \cdot (1-\theta) \cdot (1-\omega)\Big) \cdot (1-\psi),$$

for pixel positions in the window belonging to the previous frame, where

$$\begin{aligned} \omega &= \mu_d(d(\mathbf{r}, t)), \\ \theta &= \mu_\Delta(\Delta(\mathbf{r'}, t', \mathbf{r}, t)), \\ \psi &= \mu_m(m(\mathbf{r}, t)). \end{aligned}$$

2.2 Second Subfilter

Additional to the first subfilter explained in Subsection 2.1 a second subfilter is applied, which is a 3D extension of the second subfilter from [8]. This filter further improves the proposed method by reducing the noise in the colour component differences and is based on the simplified assumption that the difference between two neighbouring pixels is approximately the same in all three colour components. The central pixel in the window is estimated by combining corrected local differences in a spatio-temporal neighbourhood of each colour band.

The same $3\times3\times2$ sliding window as in the first subfilter is used (Fig. 1), which is again moved through the frame, each time filtering the central window pixel at position $(\mathbf{r}, t)$. For the filtering of this central pixel, local differences (gradients) are calculated in each colour band separately and for each pixel in the sliding window. For the red, green and blue colour band these differences are denoted by respectively LD_R, LD_G and LD_B.

For the window pixels belonging to the current frame ($t' = t$), the differences are calculated on the output of the first subfilter:

$$\begin{aligned} LD_R(\mathbf{r'}, t) &= I_f(\mathbf{r'}, t, 1) - I_f(\mathbf{r}, t, 1), \\ LD_G(\mathbf{r'}, t) &= I_f(\mathbf{r'}, t, 2) - I_f(\mathbf{r}, t, 2), \\ LD_B(\mathbf{r'}, t) &= I_f(\mathbf{r'}, t, 3) - I_f(\mathbf{r}, t, 3). \end{aligned}$$

For the window pixels belonging to the previous frame ($t' = t-1$), we use the already present output of the second subfilter, denoted by *Out*:

$$\begin{aligned} LD_R(\mathbf{r'}, t') &= Out(\mathbf{r'}, t', 1) - I_f(\mathbf{r}, t, 1), \\ LD_G(\mathbf{r'}, t') &= Out(\mathbf{r'}, t', 2) - I_f(\mathbf{r}, t, 2), \\ LD_B(\mathbf{r'}, t') &= Out(\mathbf{r'}, t', 3) - I_f(\mathbf{r}, t, 3). \end{aligned}$$

These local differences are used to calculate one correction term $\epsilon(\mathbf{r'}, t')$ for each pixel in the filtering window. This correction term is the arithmetic mean of the local differences in the red, green and blue component of the considered pixel:

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

To avoid the influence of pixels from the previous frame, belonging to another object, each pixel in the window is given a weight $WT^{\mathbf{r},t}_{\mathbf{r'},t'}$, based on the motion value $m(\mathbf{r}, t)$ already computed in the first subfilter:

$$WT^{\mathbf{r},t}_{\mathbf{r'},t'} = \begin{cases} 1, & t' = t \\ 1 - m(\mathbf{r}, t), & t' = t-1 \end{cases}.$$

The output of the second subfilter for the central window pixel is then determined as:

$$Out(\mathbf{r},t,1) = \frac{\sum_{\mathbf{r'}} WT^{\mathbf{r},t}_{\mathbf{r'},t}\Big(I_f(\mathbf{r'},t,1) - \epsilon(\mathbf{r'},t)\Big)}{\sum_{\mathbf{r'}}\sum_{t'} WT^{\mathbf{r},t}_{\mathbf{r'},t'}} + \frac{\sum_{\mathbf{r'}} WT^{\mathbf{r},t}_{\mathbf{r'},t-1}\Big(Out(\mathbf{r'},t-1,1) - \epsilon(\mathbf{r'},t-1)\Big)}{\sum_{\mathbf{r'}}\sum_{t'} WT^{\mathbf{r},t}_{\mathbf{r'},t'}},$$

where $\epsilon(\mathbf{r'},t')$ is the correction term for the components of the neighbouring pixel at position $(\mathbf{r'},t')$ and $Out(\mathbf{r},t,1)$ is the red component of the output. The green $(Out(\mathbf{r},t,2))$ and blue $(Out(\mathbf{r},t,3))$ colour component of the output are determined analogously.

3 Experimental Results

In this section some experimental results are presented. In these experiments we have applied the proposed colour video denoising technique on four different test sequences ("Salesman", "Tennis", "Chair" and "Flowers") corrupted with additive Gaussian noise of zero mean and standard deviation $\sigma_n = 5, 10, 15, 20, 25$.

As measures of objective similarity and dissimilarity between the original and the filtered frames, the peak signal to noise ratio (PSNR) and the normalized colour difference (NCD) are used. The PSNR value for colour images modelled in the RGB colour space is defined as:

$$MSE(I_0, I_f) = \frac{\displaystyle\sum_{c=1}^{3}\sum_{i=1}^{m}\sum_{j=1}^{n}(I_o(i,j,c) - I_f(i,j,c))^2}{3\cdot n\cdot m},$$

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

where $I_o(i,j,1)$, $I_o(i,j,2)$, $I_o(i,j,3)$ and $I_f(i,j,1)$, $I_f(i,j,2)$, $I_f(i,j,3)$ respectively denote the red, green and blue component of the pixel at spatial position (i,j) in respectively the original and the filtered frame, each containing m rows and n columns of pixel positions. S denotes the maximum possible value of a pixel component (here $S = 255$). 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 still stand for the original and the filtered frame respectively,

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

and

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

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

We have compared the proposed filtering framework (FMDAF-RGB-vector) to the filtering approach that denoises the Y component of the YUV transform with the wavelet extension of the original greyscale method [6] (which outperforms other greyscale methods of a similar complexity as shown in [6]) combined with an additional averaging (3×3 window) of the chrominance components U and V (FMDAF-YUV). Tables 2 and 3 show for different noise levels the average PSNR and NCD values for the test sequences processed by the two approaches. It can be concluded that, both in terms of PSNR and NCD, the proposed filtering framework is a better alternative for the usually applied filtering of the Y-component. This conclusion is however not fully confirmed visually. The original and noisy ($\sigma_n = 10$) "Salesman" sequence together with the filtered result obtained by respectively the FMDAF-RGB-vector and the FMDAF-YUV method are available on `http://users.ugent.be/~tmelange/IPMU`. When looking carefully to e.g. the left side of the phone, we see that some red and green shine (colour artefacts) is visible in the result of the FMDAF-YUV method. This is much less the case in the result of the FMDAF-RGB-vector method, which might explain the good PSNR and NCD values.

Table 2: The average PSNR value for the processed test sequences.

Sequence	**noise level**	FMDAF-RGB-vector	WRFMDAF-YUV
"Salesman"	5	38.29	37.37
	10	34.44	33.64
	15	32.20	31.20
	20	30.61	29.35
	25	29.38	27.88
"Tennis"	5	34.44	33.40
	10	31.92	29.83
	15	30.08	27.92
	20	28.71	26.68
	25	27.59	25.59
"Chair"	5	40.22	39.92
	10	36.41	35.67
	15	34.27	32.99
	20	32.63	30.89
	25	31.24	29.16
"Flower garden"	5	29.72	29.03
	10	28.38	26.39
	15	26.93	24.98
	20	25.57	23.67
	25	24.37	22.44

Table 3: The average NCD value for the processed test sequences.

Sequence	**noise level**	FMDAF-RGB-vector	WRFMDAF-YUV
"Salesman"	5	0.0450	0.0531
	10	0.0607	0.0829
	15	0.0736	0.1121
	20	0.0861	0.1397
	25	0.0985	0.1652
"Tennis"	5	0.0319	0.0339
	10	0.0393	0.0475
	15	0.0457	0.0601
	20	0.0516	0.0724
	25	0.0576	0.0848
"Chair"	5	0.0112	0.0122
	10	0.0164	0.0215
	15	0.0205	0.0309
	20	0.0243	0.0404
	25	0.0281	0.0498
"Flower garden"	5	0.0774	0.0792
	10	0.0858	0.0945
	15	0.0952	0.1064
	20	0.1046	0.1190
	25	0.1138	0.1319

We see however also that the wavelet domain method has removed more noise and produces a smoother result. Further, it should also be noticed that in an average over the frames of PSNR and NCD values, temporal discontinuities between successive frames are not taken into account. It can be concluded that better objective measures are needed for colour images and image sequences.

4 Conclusion

In this paper we have presented a new fuzzy video filter for the removal of white gaussian noise in colour image sequences. We have extended the fuzzy logic framework from our previous work [6] to colour videos through a vector based approach and combined it with a 3D extension of the colour restorating subfilter from [8].

Experimental results show that the proposed colour video denoising method performs very well in terms of average PSNR and NCD. It should however also be concluded that better objective measures are needed for colour images and image sequences.

Acknowledgement. This research was financially supported by the FWO project G.0667.06 of Ghent University.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] Mélange, T., Zlokolica, V., Schulte, S., De Witte, V., Nachtegael, M., Pizurica, A., Kerre, E.E., Philips, W.: "Video denoising by fuzzy motion and detail adaptive averaving", *Submitted to the SPIE Journal of Electronic Imaging.*

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.