A Fuzzy Filter for the Removal of Gaussian Noise in Colour Image Sequences

Tom Mélange, Mike Nachtegael, Etienne E. Kerre

Ghent University, Dept. of Applied Math. and Computer Science
Fuzziness and Uncertainty Modelling Research Group
Krijgslaan 281-S9, 9000 Ghent, Belgium
Email: {tom.melange,mike.nachtegael,etienne.kerre}@ugent.be

Abstract— *In this paper a new fuzzy filter for the removal of additive Gaussian noise in colour image sequences is presented. The proposed method, which is a colour extension of our previous work on greyscale images, consists of two subfilters. The first subfilter averages the noise in each of the colour bands separately. In this averaging, the weights of the neighbouring pixels are determined by fuzzy rules. Even though the filtering is performed in each colour band separately, information from the other colour bands is also used in the fuzzy rules. In this way, we can be more confident whether a neighbouring pixel is similar to the pixel being filtered. In order to further improve the results, a second subfilter is applied too. This colour restorating filter is based on the simplified assumption that for similar pixels the pixel value difference should be approximately the same in all colour bands. A pixel is estimated from its neighbours by estimating the differences in each colour band equal to the average over all colour bands. Experimental results show that, in terms of average PSNR and average NCD, the proposed colour extension of our previous work outperforms the usual colour extension of a greyscale filter, in which the Y component in the YUV transform is filtered with the original greyscale method and the chrominance bands U and V are averaged.*

Keywords— Video, noise filter, colour, Gaussian noise, fuzzy rule.

1 Introduction

Images sequences are among the most important information carriers in today's world. They are used in numerous applications such as broadcasting, video-phone, traffic observations, surveillance systems and autonomous navigation to name a few. The used sequences are however often affected by noise due to bad acquisition, transmission or recording. In this paper we will concentrate on image sequences corrupted with additive white Gaussian noise of zero mean and variance σ^2:

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

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 goal of noise filtering is not only visual improvement but also an improvement of the further analysis or coding of the sequences. During this filtering process a compromise needs to found between the removal of noise and the preservation of fine image details.

Most video filters that exist in literature are designed for greyscale sequences. Some examples are e.g. [2, 3, 4, 5]. These greyscale methods can nevertheless be extended to the RGB colour space [1] in a straightforward way by filtering each of the colour bands R, G and B separately. This might however result in the introduction of colour artefacts since the correlation between the different colour bands is neglected. Therefore, the commonly used alternative is to filter only the luminance component Y of the YUV-transform with the given greyscale method, possibly with an additional averaging of the chrominance components U and V.

The proposed filtering framework can be seen as a colour extension of our previous work presented in [4], which was inspired by the multiple class averaging filter from [5]. The presented filter consists of two separate subfilters. In the first subfilter, we add colour information to the fuzzy logic framework from our work in [4]. In this first fuzzy subfilter, each of the colour bands is denoised separately by averaging the noise. The weights assigned to the pixels considered in the averaging are determined by fuzzy rules. Even though the filtering is performed in each colour band separately, the fuzzy rules also require information from the other colour bands. Due to this increase in information, we can expect a more reliable estimation of the degree to which a neighbouring pixel is similar to the pixel that is filtered. However, especially around edges in the image, some colour artefacts might appear because sometimes not enough similar neighbours can be found to completely average the noise and it might also happen that a neighbouring pixel is wrongly considered as belonging to the same object (similar). To cope with this problem, the first subfilter is combined with an additional second subfilter, which is an extension of the second subfilter in [6]. This subfilter is based on the simplified assumption that for similar pixels the pixel value differences in the three different colour components should all three be approximately the same. The pixel being filtered is estimated from a neighbour by estimating the differences in each band equal to the average over the different colour bands.

The experimental results show that the proposed colour video denoising framework performs better than the YUV-colour extension in terms of average PSNR and NCD.

The paper is structured as follows: Section 2 gives some preliminaries about fuzzy set theory. Next, the proposed colour video denoising algorithm is explained in Section 3. Finally, experimental results are presented in Section 4.

2 Preliminaries

In this section, we will introduce some basic notions concerning fuzzy set theory and fuzzy if-then rules.

2.1 Fuzzy Sets

In classical set theory, an element x in a universe X belongs or does not belong to a certain set C, defined over the uni-

ISBN: 978-989-95079-6-8

verse X. This can be modelled by membership degrees belonging to $\{0,1\}$. In fuzzy set theory [7], there is a more gradual transition between 'belonging to' and 'not belonging to'. A fuzzy set F in the universe X is characterized by a $X \rightarrow [0,1]$ mapping μ_F, which assigns a degree of membership $\mu_F(x) \in [0,1]$ of x in the fuzzy set F, with every element x in X. This membership degree may also lie between 0 and 1, which makes fuzzy sets very useful and more natural than crisp sets to work with complex systems and human knowledge, where linguistic variables are used. In this paper, we will use a linguistic variable "large" for several parameters that will be introduced further. Using fuzzy set theory, a parameter is not necessarily large or not large, but can also be large to some degree.

2.2 Fuzzy Rules and Fuzzy Logical Operators

The general form of a fuzzy rule is "IF A THEN B", where the premise A and the consequent B are (collections of) propositions containing linguistic variables. These propositions can contain AND, OR and NOT operators, which correspond to respectively the intersection, the union and the complement of fuzzy sets.

The membership degree of an element x in the intersection (respectively union) of two fuzzy sets A and B in X is obtained through a mapping T, (respectively S) that maps the membership degrees of an element in the fuzzy sets A and B onto its membership degree in the fuzzy set $A \cap B$ (respectively $A \cup B$): $\mu_{(A \cap B)}(x) = T(\mu_A(x), \mu_B(x))$, $\forall x \in X$ (respectively $\mu_{(A \cup B)}(x) = S(\mu_A(x), \mu_B(x))$,$\forall x \in X$). In fuzzy logic for the mappings T and S, respectively a triangular norm [8] and a triangular conorm [8] are used. For the results in this paper, we have used the algebraic product and the probabilistic sum as triangular norm and conorm respectively. This norm and conorm led to the best results for our filter and are the most simple from a computational point of view.

To specify the complement of a fuzzy set A in X, we use a mapping N to derive the membership degree of an element x in the complement of the fuzzy set A from its membership degree in the fuzzy set A: $\mu_{coA}(x) = N(\mu_A(x))$, $\forall x \in X$. In fuzzy logic for the mapping N an involutive negator [8] is used. In this paper we have chosen for the well-known standard negator $N_s(x) = 1 - x, \forall x \in [0,1]$.

Let's take for example 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, which corresponds to the activation degree of the rule, i.e. the degree of thruthfulness, is then calculated as:

$$\mu_Z(z) = (\mu_U(u) \cdot \mu_V(v)) + (1 - \mu_W(w)) - (\mu_U(u) \cdot \mu_V(v)) \cdot (1 - \mu_W(w)). \quad (2)$$

3 The proposed algorithm

In this section, we will outline the proposed filtering framework. The method is a superposition of two subfilters, presented in respectively Subsection 3.1 and Subsection 3.2.

In the first subfilter a $3 \times 3 \times 2$ sliding window is used, which is moved through the frame from top left to bottom right, each time filtering the central position in the window. This window consists of 3×3 pixels in the current frame and 3×3 pixels in the previous frame as shown in Fig. 1. The central position in the window is denoted by $(\mathbf{r}, t)$, where $\mathbf{r} = (x, y)$ and t respectively stand for the spatial and temporal position in the image sequence. An arbitrary pixel position in the sliding window (which may also be the central position) is denoted by $(\mathbf{r'}, t')$, with $\mathbf{r'} = (x + k, y + l)$, $(-1 \leq k, l \leq 1)$ and $t' = t$ or $t' = t - 1$. Further, the second subfilter uses a 3×3 window in the current frame for which similar notations will be used as for the $3 \times 3 \times 2$ window.

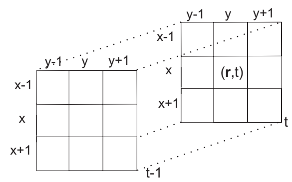

Figure 1: The $3 \times 3 \times 2$ filtering window consisting of 3×3 pixels in the current frame and 3×3 pixels in the previous frame.

3.1 First Subfilter

The subfilter explained in this subsection, can be seen as a colour extension of the filtering framework of our previous work, presented in [4]. This method was inspired by the multiple class averaging filter [5], from which we adopted the following ideas: (i) temporal blur should be avoided by taking into account only pixels from the current frame when motion is detected; (ii) image details should be preserved by filtering less when large spatial activity (details) is detected. More noise will be left, but the eye is less sensitive for the high spatial frequencies to which large spatial activity corresponds [9]. In homogeneous areas on the other hand, as much noise as possible should be removed by strong filtering.

The filtering is based on averaging the noise using the pixel values in the neighbourhood that are similar to the given pixel value and probably belong to the same object. Each colour band is filtered separately, but in the filtering of each colour band, the information from the other colour bands is used to confirm that a neighbouring pixel does indeed belong to the same object.

In the following the output of the first fuzzy subfilter is denoted by I_f, while the noisy input sequence is denoted by I_n. The output of the first subfilter for the central pixel in the window is determined as a weighted mean of the pixel values in the $3 \times 3 \times 2$ window ($i = 1, 2, 3$):

$$I_f(\mathbf{r}, t, i) = \frac{\sum_{\mathbf{r'}} \sum_{t'=t-1}^{t} W(\mathbf{r'}, t', \mathbf{r}, t, i) I_n(\mathbf{r'}, t', i)}{\sum_{\mathbf{r'}} \sum_{t'=t-1}^{t} W(\mathbf{r'}, t', \mathbf{r}, t, i)}, \quad (3)$$

The weights $W(\mathbf{r'}, t', \mathbf{r}, t, i)$ in the above weighted means are determined using fuzzy logic. The weight of a pixel is chosen equal to its membership degree in the fuzzy set "large weight". This membership degree is calculated as the activation degree of a fuzzy rule that corresponds to the ideas given at the beginning of this subsection. If large detail is detected, i.e., if a calculated detail value is large, then we should filter less by averaging only over pixels that are quite similar, i.e.,

ISBN: 978-989-95079-6-8

for which there is no large difference in the considered colour component and also in at least one of the other components. On the other hand, if there is not much detail detected, i.e., in the case that the calculated detail value is not large, strong filtering should be performed, i.e., we don't put a condition on the difference between the considered and the filtered pixel in the considered colour band (the check in the other colour bands remains for the case that the calculated detail value was not completely reliable). Further, if the pixel for which the weight is calculated belongs to the previous frame, we only want to give it a large weight if there is no motion detected in the filtering window, i.e., if a calculated motion value is not large. In the filtering of the red colour band, this results in the following two fuzzy rules, depending on whether the pixel lies in the current or the previous frame:

Fuzzy Rule 2. *Assigning the membership degree in the fuzzy set "large weight" of the weight $W(\boldsymbol{r'}, t', \boldsymbol{r}, t, 1)$ for the red value at position* **r'** *in the current frame ($t' = t$) of the window with central pixel position $(\boldsymbol{r}, t)$:*

IF $\Big($($the detail value $d(\boldsymbol{r}, t)$ *is* LARGE AND $\Delta_1(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE AND ($\Delta_2(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE OR $\Delta_3(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE))

OR ((*the detail value* $d(\boldsymbol{r}, t)$ *is* NOT LARGE) AND ($\Delta_2(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE OR $\Delta_3(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE)$\Big)$

THEN *the red value at position* **r'** *has a* LARGE *weight* $W(\boldsymbol{r'}, t', \boldsymbol{r}, t, 1)$ *in (3).*

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

IF $\Big($(*the detail value* $d(\boldsymbol{r}, t)$ *is* LARGE AND $\Delta_1(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE AND ($\Delta_2(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE OR $\Delta_2(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE))

OR ((*the detail value* $d(\boldsymbol{r}, t)$ *is* NOT LARGE) AND ($\Delta_3(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE OR $\Delta_2(\boldsymbol{r'}, t', \boldsymbol{r}, t)$ *is* NOT LARGE)$\Big)$

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

THEN *the red value at position* **r'** *has a* LARGE *weight* $W(\boldsymbol{r'}, t', \boldsymbol{r}, t, 1)$ *in (3).*

Similar fuzzy rules, switching the role of the red colour band and the colour band that needs to be filtered, are used to determine the weights $W(\mathbf{r'}, t', \mathbf{r}, t, 2)$ and $W(\mathbf{r'}, t', \mathbf{r}, t, 3)$ in expression (3) to filter the green and blue colour band respectively. In these fuzzy rules, a detail value $d(\mathbf{r}, t)$, three difference values $\Delta_1(\mathbf{r'}, t', \mathbf{r}, t)$, $\Delta_2(\mathbf{r'}, t', \mathbf{r}, t)$ and $\Delta_3(\mathbf{r'}, t', \mathbf{r}, t)$ (one for each colour band) and a motion value $m(\mathbf{r}, t)$ are used, which we will now discuss. In our proposed method, only one detail value $d(\mathbf{r}, t)$ is used for all three colour bands. This detail value depends however on three detail values computed on each colour band separately. These three detail values are equal to the standard deviation calculated in the respective colour bands on the 3×3 pixels of the filtering window belonging to the current frame ($i = 1, 2, 3$):

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

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

We don't need to know the exact value of $d(\mathbf{r}, t)$. Only the membership degree $\mu_d\big(d(\mathbf{r}, t)\big)$ of $d(\mathbf{r}, t)$ in the fuzzy set "large detail value" is needed to calculate the activation degree of Fuzzy Rules 2 and 3 that determine the weights in the expression (3). This membership degree corresponds to the activation degree of the following fuzzy rule:

Fuzzy Rule 4. *Assigning the membership degree in the fuzzy set "large detail value" of the detail value $d(\boldsymbol{r}, t)$ for the pixel at the central position $(\boldsymbol{r}, t)$ in the filtering window of the current step:*

IF $d_1(\boldsymbol{r}, t)$ *is* LARGE AND $d_2(\boldsymbol{r}, t)$ *is* LARGE AND $d_3(\boldsymbol{r}, t)$ *is* LARGE

THEN $d(\boldsymbol{r}, t)$ *is* LARGE.

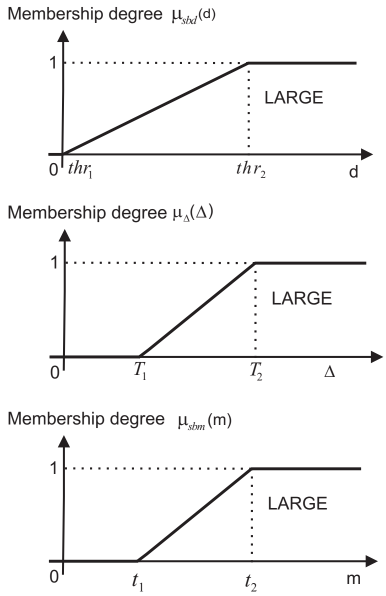

Figure 2: The respective membership functions μ_{sbd}, μ_Δ and μ_{sbm} of the respective fuzzy sets "large single band detail value", "large difference" and "large single band motion value".

The membership function μ_{sbd} of the fuzzy set "large single band detail value" is given in Fig. 2. The membership degree $\mu_d\big(d(\mathbf{r}, t)\big)$ is thus given by:

$$\mu_d\big(d(\mathbf{r}, t)\big) = \mu_{sbd}\big(d_1(\mathbf{r}, t)\big) \cdot \mu_{sbd}\big(d_2(\mathbf{r}, t)\big) \cdot \mu_{sbd}\big(d_3(\mathbf{r}, t)\big). \quad (4)$$

The three difference values $\Delta_1(\mathbf{r'}, t', \mathbf{r}, t)$, $\Delta_2(\mathbf{r'}, t', \mathbf{r}, t)$ and $\Delta_3(\mathbf{r'}, t', \mathbf{r}, t)$ that are used to determine the weights in (3) are given by ($i = 1, 2, 3$):

$$\Delta_i(\mathbf{r'}, t', \mathbf{r}, t) = |I_n(\mathbf{r'}, t', i) - I_n(\mathbf{r}, t, i)|. \quad (5)$$

The membership function μ_Δ of the fuzzy set 'large difference' is given in Fig.2.

ISBN: 978-989-95079-6-8

Analogously to the detail value $d(\mathbf{r},t)$ also only one motion value $m(\mathbf{r},t)$ is used for the filtering of all three colour bands. This value depends however again on three values computed for each of the colour bands separately. These three single band motion values are ($i = 1,2,3$):

$$\begin{aligned} m_i(\mathbf{r},t) &= \Big|I_{av}(\mathbf{r},t,i) - I_{av}(\mathbf{r},t-1,i)\Big| \\ &= \Big|\frac{1}{9}\sum_{\mathbf{r'}} I_n(\mathbf{r'},t,i) - \frac{1}{9}\sum_{\mathbf{r'}} I_v(\mathbf{r'},t-1,i)\Big|. \end{aligned}$$

Just as it was the case for the detail value $d(\mathbf{r},t)$, we don't need to know the exact value of $m(\mathbf{r},t)$ either. Only its membership degree $\mu_m\big(m(\mathbf{r},t)\big)$ in the fuzzy set "large motion value" is needed for the Fuzzy Rules 2 and 3 (or thus the calculation of the weights in (3)). This membership degree is obtained from the following fuzzy rule:

Fuzzy Rule 5. *Assigning the membership degree in the fuzzy set "large motion value" of the motion value $m(\boldsymbol{r},t)$ for the pixel at the central position $(\boldsymbol{r},t)$ in the filtering window of the current step:*

IF $(m_1(\boldsymbol{r},t)$ *is* LARGE AND $m_2(\boldsymbol{r},t)$ *is* LARGE) OR

$(m_1(\boldsymbol{r},t)$ *is* LARGE AND $m_3(\boldsymbol{r},t)$ *is* LARGE) OR

$(m_2(\boldsymbol{r},t)$ *is* LARGE AND $m_3(\boldsymbol{r},t)$ *is* LARGE)

THEN $m(\boldsymbol{r},t)$ *is* LARGE.

The membership function μ_{sbm} of the fuzzy set "large single band motion value" is given in Fig 2. The membership degree $\mu_m\big(m(\mathbf{r},t)\big)$ is thus given by:

$$\mu_m\big(m(\mathbf{r},t)\big) = \alpha + (\beta+\gamma-\beta\cdot\gamma) - \alpha\cdot(\beta+\gamma-\beta\cdot\gamma), \quad (6)$$

with

$$\begin{aligned} \alpha &= \Big(\mu_{sbm}\big(m_1(\mathbf{r},t)\big)\cdot\mu_{sbm}\big(m_2(\mathbf{r},t)\big)\Big), \\ \beta &= \Big(\mu_{sbm}\big(m_1(\mathbf{r},t)\big)\cdot\mu_{sbm}\big(m_3(\mathbf{r},t)\big)\Big), \\ \gamma &= \Big(\mu_{sbm}\big(m_2(\mathbf{r},t)\big)\cdot\mu_{sbm}\big(m_3(\mathbf{r},t)\big)\Big). \end{aligned} \quad (7)$$

Summarized, the membership degree of the pixel at position $(\mathbf{r'},t')$ in the fuzzy set "large weight", which corresponds to the weight $W(\mathbf{r'},t',\mathbf{r},t,1)$ in (3) is thus given by

$$W(\mathbf{r'},t',\mathbf{r},t,1) = \omega\cdot\theta\cdot\phi + (1-\omega)\cdot\phi - (\omega\cdot\theta\cdot\phi)\cdot((1-\omega)\cdot\phi), \quad (8)$$

where

$$\begin{aligned} &\omega = \mu_d(d(\mathbf{r},t)) \\ &\theta = (1-\mu_\Delta(\Delta_1(\mathbf{r'},t',\mathbf{r},t))) \\ &\phi = (1-\mu_\Delta(\Delta_2(\mathbf{r'},t',\mathbf{r},t))) + (1-\mu_\Delta(\Delta_3(\mathbf{r'},t',\mathbf{r},t))) \\ &\quad - (1-\mu_\Delta(\Delta_2(\mathbf{r'},t',\mathbf{r},t)))\cdot(1-\mu_\Delta(\Delta_3(\mathbf{r'},t',\mathbf{r},t))) \end{aligned}$$

For pixel positions in the window belonging to the current frame. For pixel positions in the window belonging to the previous frame, an extra factor $1-\mu_m(m(\mathbf{r},t))$ is needed.

3.2 Second Subfilter

Because sometimes not enough similar neighbours can be found to completely average the noise in the first subfilter and because some pixels might have been wrongly considered similar in the first subfilter, some colour artefacts might still be present after applying the first subfilter. To further improve the result, the first subfilter is combined with an additional second subfilter, which is an extension of the second subfilter in [6]. Based on the simplified assumption that the difference between similar pixels is approximately the same in all three colour bands, a pixel is estimated from a neighbour by estimating a difference in a given colour component equal to the average over all three colour bands. So a difference that is larger than the average is made smaller and vice versa. The final output is a weighted average over the estimations obtained from the different neighbours, where the weight is the degree to which we believe that the neighbour belongs to the same object. The weights are introduced because for neighbours not belonging to the same object, the simplified assumption does not hold.

3.2.1 Local Differences and Correction Terms

As mentioned before, for this second subfilter, a 3×3 sliding window 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 and they are calculated based on the output of the first subfilter ($i = 1,2,3$):

$$LD_i(\mathbf{r'},t,\mathbf{r},t) = I_f(\mathbf{r'},t,i) \; - \; I_f(\mathbf{r},t,i). \quad (9)$$

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

$$\epsilon(\mathbf{r'},t,\mathbf{r},t) = \frac{1}{3}\sum_{i=1}^{3} LD_i(\mathbf{r'},t,\mathbf{r},t). \quad (10)$$

3.2.2 Output of the second subfilter

In [6], the output for each component of the central pixel is an average of the corresponding components of the neighbourhood pixels, corrected with the corresponding correction term ($i = 1,2,3$):

$$Out(\mathbf{r},t,i) = \frac{\sum_{\mathbf{r'}}\Big(I_f(\mathbf{r'},t,i) - \epsilon(\mathbf{r'},t,\mathbf{r},t)\Big)}{9}. \quad (11)$$

However, pixels that belong to another object and that have another colour, have a negative influence on the output. In homogeneous areas, neighbouring pixels are expected to be almost the same, and the local differences to be almost 0. So the method further averages the remaining differences caused by the noise. For a pixel belonging to another object however, the assumption that the local differences are expected to be equal in all components does not always hold. Therefore we assign weights $WT(\mathbf{r'},t,\mathbf{r},t)$ to the neighbouring pixels, based on whether they are expected to belong to the same object or not. To make this decision, we use the Euclidian distance between the central pixel and the considered neighbourhood pixel, given by

$$\delta(\mathbf{r'},t,\mathbf{r},t) = \big(\sum_{i=1}^{3} LD_i(\mathbf{r'},t,\mathbf{r},t)^2\big)^{\frac{1}{2}} \quad (12)$$

ISBN: 978-989-95079-6-8

The weights themselves are then calculated using the following fuzzy rule that expresses that the value $\delta(\mathbf{r'}, t, \mathbf{r}, t)$ should not be large. Otherwise, the considered pixel is expected to belong to another object.

Fuzzy Rule 6. *Assigning the weight in the second subfilter for the pixel at position* $(\mathbf{r'}, t)$ *in the filtering window:*

IF $\delta(\mathbf{r'}, t, \mathbf{r}, t)$ *is* NOT LARGE

THEN *the pixel at position* $(\mathbf{r'}, t)$ *has a* LARGE WEIGHT $WT(\mathbf{r'}, t, \mathbf{r}, t)$ *in the second subfilter.*

The membership function μ_δ that determines the fuzzy set "large Euclidian distance" is depicted in Fig 3. The weights in

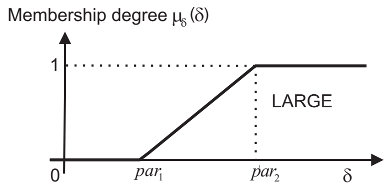

Figure 3: The membership function μ_δ of the fuzzy set "large Euclidian distance".

the filtering are again chosen equal to their membership degree in the fuzzy set "large weight", i.e., $WT(\mathbf{r'}, t, \mathbf{r}, t) = 1 - \mu_\delta(\delta(\mathbf{r'}, t, \mathbf{r}, t))$.

Finally, if not $WT(\mathbf{r'}, t', \mathbf{r}, t) = 0$ for all neighbouring pixels in the 3×3 window, the output of the second subfilter for the central pixel in the window is determined as follows ($i = 1, 2, 3$):

$$Out(\mathbf{r}, t, i) = \frac{\sum_{\mathbf{r'}} WT(\mathbf{r'}, t, \mathbf{r}, t)\Big(I_f(\mathbf{r'}, t, i) - \epsilon(\mathbf{r'}, t, \mathbf{r}, t)\Big)}{\sum_{\mathbf{r'}} WT(\mathbf{r'}, t', \mathbf{r}, t)} \quad (13)$$

where $\epsilon(\mathbf{r'}, t, \mathbf{r}, t)$ is the correction term for the components of the neighbouring pixel at position $(\mathbf{r'}, t')$. If the central pixel is so corrupt that all neighbouring pixels get a weight equal to zero, the output is calculated by giving all neigbhouring pixels in the window a weight equal to 1 and the corrupt central pixel the weight 0 ($i = 1, 2, 3$):

$$Out(\mathbf{r}, t, i) = \frac{\sum_{\mathbf{r'} \neq \mathbf{r}} \Big(I_f(\mathbf{r'}, t, i) - \epsilon(\mathbf{r'}, t, \mathbf{r}, t)\Big)}{8} \quad (14)$$

4 Experimental Results

In this section we present some experimental results obtained from the test sequences "Salesman", "Tennis", "Chair" and "Flowers", corrupted with additive Gaussian noise of zero mean and standard deviation $\sigma_n = 5, 10, 15, 20, 25$.

To draw conlusions about the proposed filtering framework, we have used the peak signal to noise ratio (PSNR) and the normalized colour difference (NCD) as measures of objective similarity and dissimilarity between the original and the filtered frames [6]. The higher the PSNR and the lower the NCD, the more similar the original and the filtered frame.

4.1 Parameter Selection

The parameters, that determine the membership functions in the above described filtering framework, have been set as follows. For the respective noise levels $\sigma_n = 5, 10, 15, 20$, the optimal parameters in terms of the mean PSNR values averaged over the sequences "Salesman", "Tennis", "Flowers" and "Chair" have been determined by letting them vary over a range of possible values. As illustrated in Fig. 4 for the parameter thr_2, this led to a linear relationship between these optimal values and the noise level. Hence, the parameters are set as the best fitting line through the observations, as shown in Fig. 4. The equations of those straight lines are given in Table 1, where σ_n stands for the standard deviation of the Gaussian noise. If this standard deviation is not known, it can be estimated by e.g. the method from [10].

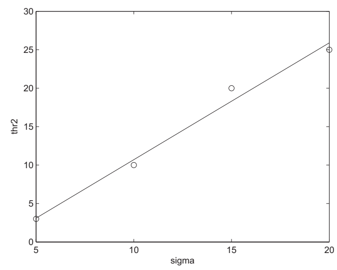

Figure 4: Optimal parameter values in terms of the PSNR.

Table 1: The used parameter values.

Parameter	Value	Parameter	Value
thr_1	0	t_1	$0.72\sigma_n - 4.0$
thr_2	$1.52\sigma_n - 4.5$	t_2	$2.22\sigma_n - 4.5$
T_1	0	par_1	$1.1\sigma_n - 7.5$
T_2	$3.14\sigma_n - 1.0$	par_2	$6\sigma_n + 35$

4.2 Experiments

In the experiments, we have compared the proposed filter (denoted by FMDAF-CR) to the filtering scheme in which a wavelet extension of the original pixel domain greyscale method [4], which outperforms both the pixel domain filter and other state-of-the-art greyscale methods of a similar complexity (as shown in [4]), is applied on the Y component of the YUV transform, followed by an averaging (3×3 window) of the chrominance components U and V (denoted by FMDAF-YUV). The average PSNR and NCD values found for the test sequences corrupted with different noise levels and processed by these two above approaches are given in Tables 2 and 3 respectively. From these tables, we can conclude that the proposed colour extension performs better in terms of average PSNR and average NCD than the commonly used YUV-filtering. For a visual comparison, we have made the original and noisy ($\sigma_n = 15$) "Salesman" sequence and the results after applying respectively the FMDAF-CR and the FMDAF-YUV filter available on `http://users.ugent.be/~tmelange/colourrule`. If

ISBN: 978-989-95079-6-8

Table 2: Average PSNR value for the processed sequences.

Sequence	σ_n	FMDAF-CR	WRFMDAF-YUV
“Salesman”	5	38.34	37.37
	10	34.36	33.64
	15	32.07	31.20
	20	30.45	29.35
	25	29.18	27.88
“Tennis”	5	36.51	33.40
	10	32.75	29.83
	15	30.35	27.92
	20	28.68	26.68
	25	27.39	25.59
“Chair”	5	39.27	39.92
	10	36.17	35.67
	15	33.94	32.99
	20	32.17	30.89
	25	30.67	29.16
“Flower garden”	5	33.32	29.03
	10	30.07	26.39
	15	27.65	24.98
	20	25.87	23.67
	25	24.47	22.44

Table 3: Average NCD value for the processed sequences.

Sequence	σ_n	FMDAF-CR	WRFMDAF-YUV
“Salesman”	5	0.0446	0.0531
	10	0.0676	0.0829
	15	0.0861	0.1121
	20	0.1027	0.1397
	25	0.1183	0.1652
“Tennis”	5	0.0269	0.0339
	10	0.0376	0.0475
	15	0.0470	0.0601
	20	0.0555	0.0724
	25	0.0638	0.0848
“Chair”	5	0.0132	0.0122
	10	0.0182	0.0215
	15	0.0238	0.0309
	20	0.0296	0.0404
	25	0.0353	0.0498
“Flower garden”	5	0.0543	0.0792
	10	0.0743	0.0945
	15	0.0907	0.1064
	20	0.1041	0.1190
	25	0.1157	0.1319

we concentrate on the side of the phone, we see that for the FMDAF-YUV method more colour artefacts (green and red spots) are visible than for the FMDAF-CR filter, which might be an explanation for the better PSNR and NCD values. Further, we also see that the wavelet domain method FMDAF-YUV has removed more noise and produces a smoother result. This smoother result can however be attributed to the use of a wavelet domain filter. If the original pixel domain greyscale method would have been used, we would also have had a little more noise remaining. Remark also that the smoother result also has as a result that the details have been smoothed a little more. This trade-off between noise removal and detail preservation is one of the main challenges in the development of a noise filter.

Acknowledgment

This work was financially supported by the GOA-project B/04138/01 IV 1 of Ghent University.

References

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

[2] E.J. Balster, Y.F. Zheng and R.L. Ewing. Combined spatial and temporal domain wavelet shrinkage algorithm for video denoising. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(2):220–230, Feb. 2006.

[3] S.M. Mahbubur Rahman, M. Omair Ahmad and M.N.S. Swamy. Video denosing based on inter-frame statistical modeling of wavelet coefficients. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(2):187–198, Feb. 2007.

[4] T. Mélange et al. A new fuzzy motion and detail adaptive video filter. *Proc. of ACIVS 2007, LNCS*, 4678:640–651, 2007.

[5] V. Zlokolica, A. Pizurica and W. Philips. Video denoising using multiple class averaging with multiresolution. *International Workshop VLBV03, LNCS,* 2849:172–179, 2003.

[6] S. Schulte et al. A New Fuzzy Additive Noise Reduction Method. *Proc. of ICIAR 2007, LNCS* 4633:12–23, 2007.

[7] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353,1965.

[8] S. Weber. A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms. *Fuzzy Sets and Systems*, 11(2):115–134, 1983.

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

[10] D. Donoho and I. Johnstone. Ideal spatial adaptation by wavelet shrinkage, *Biometrika*, 8:425–455, 1994.

ISBN: 978-989-95079-6-8