EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Robust fuzzy scheme for Gaussian denoising of 3D color video

Alberto Jorge Rosales-Silva[*], Francisco Javier Gallegos-Funes, Ivonne Bazan Trujillo and Alfredo Ramírez García

Abstract

We propose a three-dimensional Gaussian denoising scheme for application to color video frames. The time is selected as a third dimension. The algorithm is developed using fuzzy rules and directional techniques. A fuzzy parameter is used for characterization of the difference among pixels, based on gradients and angle of deviations, as well as for motion detection and noise estimation. By using only two frames of a video sequence, it is possible to efficiently decrease Gaussian noise. This filter uses a noise estimator that is spatio-temporally adapted in a local manner, in a novel way using techniques mentioned herein, and proposing a fuzzy methodology that enhances capabilities in noise suppression when compared to other methods employed. We provide simulation results that show the effectiveness of the novel color video denoising algorithm.

Keywords: Fuzzy theory; Video frames; Motion detection; Directional processing; Gaussian noise

1. Introduction

All pixels in digital color video frames are commonly affected by Gaussian-type noise due to the behavior of the image acquisition sensor; in accordance with this, we make the following assumptions as to the noise:

$$G(x_\beta) = N(0,\sigma) = \frac{1}{\sqrt{2\pi}\cdot\sigma}\exp\left(\frac{-x_\beta{}^2}{2\cdot\sigma^2}\right), \tag{1}$$

where x_β represents the original pixel component value, β = {*Red*, *Green*, *Blue*} are the notations on each pixel color component (or channel), and σ is the standard deviation of the noise. In our case, the Gaussian function is independently used on the pixel component of each channel of the frame in order to obtain the corrupted video sequence.

A pre-processing procedure to reduce noise effect is the main stage of any computer vision application. It should include procedures to reduce the noise impact in a video without degrading the quality, edges, fine detail, and color properties.

The current proposal is an attempt to enhance the quality while processing the color video sequences corrupted by Gaussian noise; *this methodology is an extension of the method proposed for impulsive noise removal* [1]. There exist numerous algorithms that perform the processing of 3D signals using only the spatial information [2]. Other applications use only the temporal information [3,4]; an example is one that uses wavelet procedures to reduce the delay in video coding [5]. There exist also some interesting applications that use spatio-temporal information [6-13]. The disadvantage of these 3D solutions is that they often require large memory and may introduce a significant time delay in cases where there is a need for more than one frame to be processed. This is undesirable in interactive applications such as infrared camera-assisted driving or videoconferencing. Moreover, full 3D techniques tend to require more computation than separable ones, and their optimal performance can be very difficult to determine. For example, integrating video coding and denoising is a novel processing paradigm and brings mutual benefits to both video processing tools. In Jovanov et al. [14], the main idea is the reuse of motion estimation resources from the video coding module for the purpose of video denoising. Some disadvantages of the work done by Dai et al. [15] is that they use a number of reference frames that increases the computational charge; the algorithm MHMCF was originally applied to grayscale video signal; and in the paper referenced [14], it was adapted to color video denoising, transforming the RGB video in a luminance color difference space proposed by the authors.

[*] Correspondence: arosaless@ipn.mx
School of Advanced Mechanical and Electrical Engineering, U.P.A.L.M, Av. Instituto Politécnico Nacional S/N, ESIME Zacatenco, Col. Lindavista, México D.F. 07738, Mexico

Other state-of-the-art algorithms found in literature work in the same manner; for example in Liu and Freeman [16], a framework that integrates robust optical flow into a non-local means framework with noise level estimation is used, and the temporal coherence is taken into account in removing structured noise. In the paper by Dabov et al. [17], it is interesting to see how they propose a method based on highly sparse signal representation in local 3D transform domain; a noisy video is processed in blockwise manner, and for each processed block, they form data array by stacking together blocks found to be similar to the currently processed one. In [18], Mairal et al. presented a framework for learning multiscale sparse representations of color images and video with overcomplete dictionaries. They propose a multiscaled learned representation obtained by using an efficient quadtree decomposition of the learned dictionary and overlapping image patches. This provides an alternative to predefined dictionaries such as wavelets.

The effectiveness of the algorithm designed is justified by comparing it with four other state-of-the-art approaches: 'Fuzzy Logic Recursive Spatio-Temporal Filter' (FLRSTF), where a fuzzy logic recursive scheme is proposed for motion detection and spatio-temporal filtering capable of dealing with Gaussian noise and unsteady illumination conditions in both the temporal and the spatial directions [19]. Another algorithm used for comparison is the 'Fuzzy Logic Recursive Spatio-Temporal Filter using Angles' (FLRSTF_ANGLE). This algorithm uses the angle deviations instead of gradients as a difference between pixels in the FLRSTF algorithm. The 'Video Generalized Vector Directional Filtering in Gaussian Denoising' (VGVDF_G) [20] is a directional technique that computes the angle deviations between pixels as a difference criterion among them. As a consequence, the vector directional filters (VDF) do not take into account the image brightness when processing the image vectors. Finally, the 'Video Median M-type K-Nearest Neighbor in Gaussian Denoising' filter (VMMKNN_G) [21,22] uses order statistics techniques to characterize the pixel differences.

The proposed algorithm employs only two frames in order to reduce the computational processing charge and memory requirements, permitting one to produce an efficient denoising framework. Additionally, it applies the relationship that the neighboring pixels have to the central one in *magnitude* and *angle deviation*, connecting them by fuzzy logic rules designed to estimate the motion and noise parameters. The effectiveness of the present approach is justified by comparing it with four state-of-the-art algorithms found in literature as explained before.

The digital video database is formed by the *Miss America*, *Flowers*, and *Chair* color video sequences; this database is well known in scientific literature [23]. Frames were manipulated to be 24 bits in depth to form true-color images with 176 × 144 pixels, in order to work with the Quarter Common Intermediate Format (QCIF). These video sequences were selected because of their different natures and textures. The database was contaminated by Gaussian noise at different levels of intensity for each channel in an independent manner. This was used to characterize the performance, permitting the justification of the robustness of the novel framework.

2. Proposed fuzzy design

The first frame of the color video sequence is processed as follows. First, the histogram and the mean value $\left(\bar{x}_\beta\right)$ for each pixel component are calculated, using a 3 × 3 processing window. Then, an angle deviation between two vectors $(\bar{x}\,\text{and}\,x_c)$ containing components in the Red, Green, and Blue channels is computed as $\theta_c = A(\bar{x}, x_c)$, where $\theta_c = \cos^{-1}\left\{\frac{\bar{x}\cdot x_c}{|\bar{x}|\cdot|x_c|}\right\}$ is the angle deviation of the mean value vector $(\bar{x})$ with respect to the central pixel vector (x_c) in a 3 × 3 processing window. Color-image processing has traditionally been approached in a componentwise manner, that is, by processing the image channels separately. These approaches fail to consider the inherent correlation that exists between the different channels, and they may result in pixel output values that are different from the input values with possible shifts in chromaticity [24]. Thus, it is desirable to employ vector approaches in color image processing to obtain the angle deviations.

The angle interval [0, 1] is used to determine the histogram. The pixel intensity takes values from 0 to 255 in each channel; the angle deviation θ_c for any given pixel with respect to another one falls within the interval $\left[0, \frac{\pi}{2}\right]$. The angle deviations outside the proposed interval ([0, 1]) are not taken into account in forming the histogram. Therefore, the noise estimator is obtained using only values inside this interval; this is to avoid the smoothness of some details and improve the criteria results.

It is common practice to normalize a histogram by dividing each of its components by the total number of pixels in the image; this is an estimate of the probability of occurrence of intensity levels in the image. Using this same principle, we propose the use of a normalized histogram based on angle deviations; this normalized histogram being an estimate of the probability of occurrence of the angle deviations between pixels. The procedure used to obtain the histogram is that of using the vectorial values: if $[(F-1)/255] \leq \theta_c \leq [F/255]$, the histogram is increased by '1' in the F position; the parameter F increases from 1 to 255; if the aforementioned condition does not hold for the range of F, the histogram remains unchanged for F, where θ_c is the angle deviation of the central pixel with respect to one of its neighboring pixel. The parameter F is proposed only to determine to which value of pixel intensity in a histogram the angle deviation belongs.

After obtaining the histogram, the probability of occurrence for each one of the elements of the histogram must be calculated. After the mean value $\mu = \sum_{j=0}^{255} j \cdot p_j$ is computed (where p_j is the probability of occurrence of each element in the histogram), the variance $\sigma_\beta^2 = \sum_{j=0}^{255} (j-\mu)^2 \cdot \left(p_j\right)$ (where j represents each element inside the histogram) and the *general standard deviation* (*SD*) $\sigma'_\beta = \sqrt{\sigma_\beta^2}$ are determined. The SD parameter is used as the noise estimator for the purpose of decreasing Gaussian noise only for the first frame of the video sequence. In this step of the algorithm, σ'_β is the same for all three channels of a color image for the general process of the Gaussian denoising algorithm, as Figure 1 indicates. The SD parameter is used to find the deviations representing the data in its distribution from the arithmetic mean. This is in order to present them more realistically when it comes to describing and interpreting them for decision-making purposes. We estimate the SD parameter of the Gaussian noise from the input video sequence only for the first frame (t = 0) and subsequently try to adapt the SD to the input video and noise changes by spatio-temporal adaptation of the noise estimator SD.

To summarize, we use the SD parameter as an estimate of the noise to be applied in the *spatial algorithm*, which will be renewed on a temporary adaptive filter in order to ultimately generate an adaptive spatio-temporal noise estimator.

2.1. Spatial algorithm

The spatial algorithm allows one to estimate the angle deviation of the neighboring pixels with respect to the central one. The results are adjusted according to the processing windows used (see Figure 2b). This methodology was developed to effectively identify uniform areas within the image to be processed with a fast processing algorithm, like the 'Mean Weighted Filtering Algorithm' described below.

The proposed fast filtering procedure is carried out using the following methodology under an '*IF-OR-THEN-ELSE*' condition: IF (θ_1ANDθ_3ANDθ_4ANDθ_6 ≥ τ_1) OR (θ_0ANDθ_2ANDθ_5ANDθ_7 ≥ τ_1) THEN the Mean Weighted Filtering Algorithm, ELSE the *Spatial Filtering Algorithm* (where τ_1 is a threshold defined as 0.1). All parameter values were proposed according to the best results of the *peak signal-to-noise ratio (PSNR*) and *mean absolute error (MAE*) criteria obtained after numerous experiments. If the Mean Weighted Filtering Algorithm [25] is selected, one processes the pixel's intensity components of the 3 × 3 window sample using Equation 2. This procedure is used because the angle deviation between the pixels is very small, which could indicate a uniform region where it is likely that there are no edges and details which may be softened. Thus, the use of the Mean Weighted Filtering Algorithm is proposed. Taking into account the fact that the relationship between a distance measure (angle deviation) is generally exponential, a sigmoidal linear membership function is suggested and a fuzzy weight

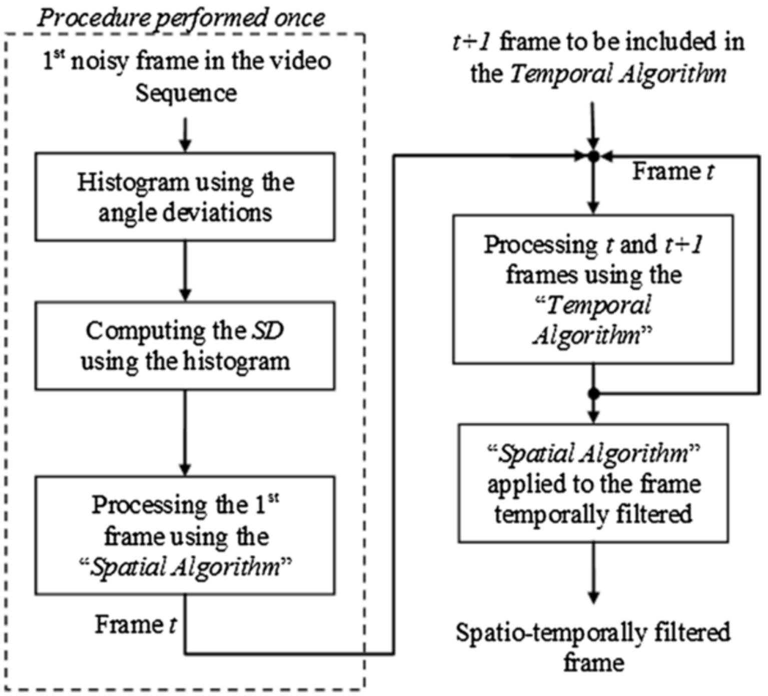

Procedure performed once
1st noisy frame in the video Sequence
Histogram using the angle deviations
Computing the SD using the histogram
Processing the 1st frame using the "Spatial Algorithm"
Frame t
t+1 frame to be included in the Temporal Algorithm
Frame t
Processing t and t+1 frames using the "Temporal Algorithm"
"Spatial Algorithm" applied to the frame temporally filtered
Spatio-temporally filtered frame

Figure 1 General scheme of the algorithm for Gaussian denoising.

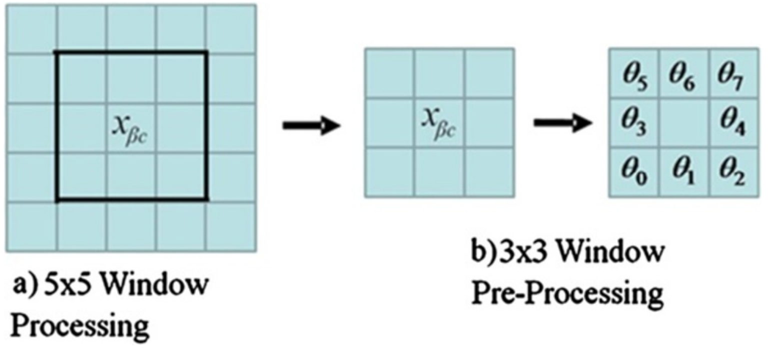


Figure 2 Processing windows used in the Gaussian denoising algorithm.

$\left(2/\left(1+e^{\theta_i}\right)\right)$ associated with the vector $x_{\beta i}$ can be used in the following equation:

$$y_{\beta\text{out}} = \left[\sum_{\substack{i=0\\ i\neq c}}^{N-1} x_{\beta i}\cdot\left(\frac{2}{\left(1+e^{\theta_i}\right)}\right)+x_{\beta c}\right] \Bigg/ \left[\sum_{i=0}^{N-1}\left[\frac{2}{\left(1+e^{\theta_i}\right)}\right]+1\right], \tag{2}$$

where $N = 8$ represents the number of data samples to be taken into account and it is in agreement with Figure 2; the fuzzy weight computed will produce an output in the interval [0,1], and it corresponds to each angle deviation value computed excluding the central angle deviation.

If the Spatial Filtering Algorithm was selected, it probably means that the sample contained edges and/or fine details. To implement this filter, the following methodology is proposed. The procedure consists of computing a new *locally adapted SD* (σ_β) for each plane of the color image, using a 5 × 5 processing window (see Figure 2a). In addition, the local updating of the SD should be undertaken according to the following condition: if $\sigma_\beta = \sigma'_\beta$, then $\sigma_\beta = \sigma'_\beta$; otherwise $\sigma'_\beta = \sigma_\beta$, where σ'_β was previously defined. This is most likely because the sample has edges and details, presenting a large value of dispersion among the pixels, so the largest SD value describes best this fact.

To provide a parameter indicating the similarity between the central and neighboring pixels, a gradient (∇) was defined. This parameter describes the magnitude differences between pixels, and the gradient is applied to each β component of the noisy color frame independently, as follows: $\nabla_{(k,l)\beta}(i, j) = |\nabla_\beta(i + k, j + l) - \nabla_\beta(i, j)| = \nabla_{\gamma\beta}$, where the pair (k, l) (with $k, l \in \{-1, 0, 1\}$) represents each of the eight cardinal directions, and (i, j) is the center of the gradient. This leads to the *main values* (see Figure 3) [19]. The eight gradient values according to the eight different directions (or neighbors) are called *main gradient values*. To provide more robustness to the algorithm, and avoid image blurring, the use of two *related gradient values* in the same direction is proposed. We assume that in case an edge like image structure extends in a certain direction $\gamma = \{N, S, E, W, SE, SW, NE, NW\}$, it leads to large derivative values perpendicular to the direction γ at the current pixel position (i, j) and at the neighboring pixels as well, in other words; these values are determined making use of a right angle with the direction of the main gradient. For example (see Figure 3a), in the SE direction (for $(k, l) = (1, 1)$ and (i, j) = (0, 0)), we calculate the main gradient value $\nabla_{(1,1)\beta}(i, j) = |\nabla_\beta(i + 1, j + 1) - \nabla_\beta(i, j)| = \nabla_{\gamma\beta M}$ and two related gradient values: $\nabla_{(2,0)\beta}(i + 1, j - 1) = |\nabla_\beta(i + 2, j) - \nabla_\beta(i + 1, j - 1)| = \nabla_{\gamma\beta D1}$ and $\nabla_{(0,2)\beta}(i - 1, j + 1) = |\nabla_\beta(i, j + 2) - \nabla_\beta(i - 1, j + 1)| = \nabla_{\gamma\beta D2}$. In such a manner, by taking into account those three derivative values for each direction (and combining them in a fuzzy logic manner), we distinguish local variations due to noise from those due to the edge-like image structures. The two *derived gradient values* are used to distinguish noisy pixels from edge pixels; when all of these gradients are larger than a predefined threshold T_β, (i, j) is considered to be a noisy pixel and must be filtered out.

Subsequently, the following condition should be verified: IF $\nabla_{\gamma\beta(M,D1,D2)} < T_\beta$ (where $T_\beta = 2\cdot\sigma_\beta$), THEN a membership value using $\nabla_{\gamma\beta(M,D1,D2)}$ is computed; otherwise, the membership value is 0. The threshold value T_β is obtained experimentally according to the PSNR and MAE criteria: γ represents each of the eight cardinal points, and $\nabla_{\gamma\beta(M,D1,D2)}$ represents each of the values computed for each of the neighboring pixels with respect to the central one, within a sliding window. These gradients are called ‘main gradient values’. Two ‘derived gradient values’ are employed, avoiding the blurring of the image in the presence of an edge instead of a noisy pixel.

The detailed methodological procedures used to compute the main and derived gradient values for the eight cardinal directions are described by Zlokolica et al. [19]. If $\nabla_{\gamma\beta(M,D1,D2)} < T_\beta$ for each of the three gradient values (the main and derived according to Figure 3a), then the angle deviation is calculated in the corresponding direction (γ). This means that if the main and derived gradient values are of a lower value than the threshold (T_β), one

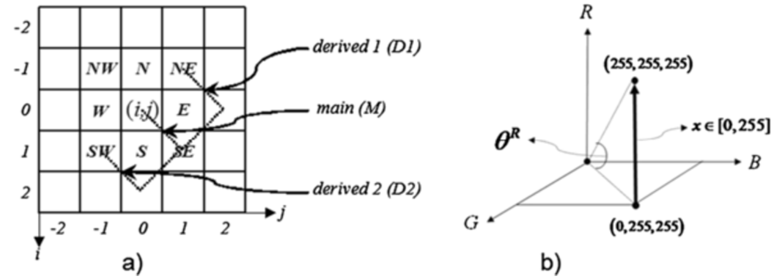


Figure 3 ***Main*** **and** ***derived*** **values (a) involved in 5 × 5 processing window; angle deviation (b) for only one plane, $\beta = R$.**

gets the angle deviations from the values for the three gradients; however, if any of these values do not satisfy the condition, the angle deviation is set to 0 for the values that do not comply.

Another way to characterize the difference between pixels is by calculating the angle deviation from the central pixel and its neighbors; this is called the *main* and *derived vectorial values*. Calculated angle deviations in the cardinal directions are taken as the weight values for each color plane of an image (Equation 3). These weights provide a relationship between pixels in a single plane at a given angle deviation. Equation 3 illustrates the calculation of the angle deviation to obtain the weight values, where

$$\theta^{\beta} = \cos^{-1}\left\{\frac{2(255)^2 + x_{\gamma\beta(M,D1,D2)}\cdot x'_{\gamma\beta(M,D1,D2)}}{\sqrt{2\cdot(255)^2 + \left(x_{\gamma\beta(M,D1,D2)}\right)^2}\cdot\sqrt{2\cdot(255)^2 + \left(x'_{\gamma\beta(M,D1,D2)}\right)^2}}\right\};$$

these values range from 0 to 1 according to Figure 3b.

$$\alpha_{\gamma\beta} = \frac{2}{\left(1 + \exp\left[\theta^{\beta}\right]\right)}, \tag{3}$$

where $x_{\gamma\beta(M,D1,D2)}$ is the pixel component in the associated direction. For example, for the $x_{\gamma\beta M}$ component of the pixel, the coordinate is (0, 0) as shown in Figure 3a. Therefore, for component $x'_{\gamma\beta M}$, the coordinate should be (1, 1) for the 'SE' cardinal direction, and so on. This parameter indicates that the smaller the difference in angle between the pixels involved, the greater the weight value of the pixel in the associated direction.

Finally, the main and derived vectorial gradient values are used to find a degree of membership using membership functions, which are functions that return a value between 0 and 1, indicating the degree of membership of an element with respect to a set (in our case, we define a BIG fuzzy set). Then, we can characterize the level of proximity of the components of the central pixel with respect to its neighbors, and see if it is a noisy or in motion component, or free of motion and/or low noise.

As mentioned above, we have defined a BIG fuzzy set; it will feature the presence of noise in the sample to be processed. The values that belong to this fuzzy set, in whole or in part, will represent the level of noise present in the pixel.

The membership function used to characterize the 'main and derived vectorial gradient values' is defined by:

$$\mu_{\mathrm{BIG}} = \begin{cases} \max\left\{\left(1-\left[\nabla_{\gamma\beta(M,D1,D2)}/T_{\beta}\right]\right), \alpha_{\gamma\beta(M,D1,D2)}\right\}, & \textit{if } \nabla_{\gamma\beta} < T_{\beta} \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

A *fuzzy rule* is created from this membership function, which is simply the application of the membership function by fuzzy operators. In this case, fuzzy operator *OR* is defined as $OR(f_1, f_2) = \max(f_1, f_2)$.

Each pixel has one returned value defined by the level of corruption present in the pixel. That is, one says 'the pixel is corrupted' if its BIG membership value is 1, and 'the pixel is low-noise corrupted' when its BIG membership value is 0. The linguistics 'the pixel is corrupted' and 'the pixel is low-noise corrupted' indicate the degree of belonging to each of the possible states in which the pixel can be found.

From the fuzzy rules, we obtain outputs, which are used to make decisions. The function defined by Equation 4 returns values between 0 and 1. It indicates how the parameter behaved with respect to the proposed fuzzy set. Finally, the following fuzzy rule is designed to connect gradient values with angle deviations, thus forming the 'fuzzy vectorial-gradient values'.

Fuzzy rule 1 helps to detect the edges and fine details using the membership values of the BIG fuzzy set obtained by Equation 4. The fuzzy values obtained by this rule are taken as fuzzy weights and used in a fast processing algorithm to improve the computational load. This fast processing algorithm is defined by means of Equation 5.

Fuzzy rule 1: the fuzzy vectorial-gradient value is defined as $\nabla_{\gamma\beta}\alpha_{\gamma\beta}$, so: IF (($\nabla_{\gamma\beta M}$, $\alpha_{\gamma\beta}$) is BIG AND ($\nabla_{\gamma\beta D1}$, $\alpha_{\gamma\beta D1}$) is BIG) OR (($\nabla_{\gamma\beta M}$, $\alpha_{\gamma\beta}$) is BIG AND ($\nabla_{\gamma\beta D2}$, $\alpha_{\gamma\beta D2}$) is BIG), THEN $\nabla_{\gamma\beta}\alpha_{\gamma\beta}$ is BIG. In this fuzzy rule, the 'AND' and 'OR' operators are defined as algebraic operations, consequently: AND = $A \cdot B$, and OR = $A + B - A \cdot B$.

The fuzzy weights are used in the fast algorithm as a final step in the noise suppression of the spatial algorithm; the fast algorithm is defined as an averaging procedure with weights as follows:

$$y_{\beta \text{out}} = \frac{\sum_{\gamma} \left(\nabla_{\gamma\beta} \alpha_{\gamma\beta} \right) \cdot x_{\gamma\beta}}{\sum_{\gamma} \left(\nabla_{\gamma\beta} \alpha_{\gamma\beta} \right)}, \tag{5}$$

where $x_{\gamma\beta}$ represents each component magnitude of the neighboring pixels around the central pixel within the pre-processing window (Figure 2b) in the respective cardinal direction, and $y_{\beta\text{out}}$ is the output of the spatial algorithm applied to the first frame of the video sequence. From this, we obtain the first spatially filtered t frame which is then passed to the temporal algorithm, joined to the t + 1 frame according to the scheme described in Figure 1.

2.2. Temporal algorithm

The outlined spatial algorithm smoothes Gaussian noise efficiently but still loses some of the image's fine details and edges. To avoid these undesirable outputs, a temporal algorithm is proposed. To design such an algorithm, only two frames of the video sequence are used. The spatially filtered t frame obtained with the methodology developed in Section 2 is used once in order to provide the temporal algorithm of a filtered t frame to be used for reference to enhance the capabilities of the temporal algorithm from the first frame of the video stream without losing significant results, and the corrupted t + 1 frame of the video sequence.

The temporal algorithm, like the spatial algorithm, is governed by fuzzy rules to help detect the noise and motion present between pixels of two frames (t and t + 1), thus avoiding the loss of important features of video frames. The proposed fuzzy rules are used for each color plane of the two frames (t and t + 1) independently. In the same way as the spatial algorithm, the gradient and the angle deviation values are calculated in order to characterize the difference between pixels in the two frames of the video sequence. These values are related to the central pixel $x_{\beta c}^{t+1}$ with respect to its neighbors in frames t and t + 1 and are computed as follows:

$$\theta_{\beta ic}^{1} = A\left(x_{\beta i}^{t}, x_{\beta c}^{t+1}\right), \ \nabla_{\beta ic}^{1} = \left| x_{\beta i}^{t} - x_{\beta c}^{t+1} \right|, \tag{6}$$
$$i, j = 0, \ldots, N; \text{where } \ N = 8,$$

$$\theta_{\beta ij}^{2} = A\left(x_{\beta i}^{t}, x_{\beta j}^{t+1}\right), \ \nabla_{\beta ij}^{2} = \left| x_{\beta i}^{t} - x_{\beta j}^{t+1} \right|, \tag{7}$$
$$\theta_{\beta jc}^{3} = A\left(x_{\beta j}^{t+1}, x_{\beta c}^{t+1}\right), \ \nabla_{\beta j}^{3} = \left| x_{\beta j}^{t+1} - x_{\beta c}^{t+1} \right|.$$

This is better understood with an example, as illustrated in Figure 4, for the case where β = Red (R), and i = j = 2.

Similarly defined as was the BIG fuzzy set, this set is defined as the SMALL fuzzy set. The same meanings for the expressions 'the pixel is corrupted' and the 'the pixel is low-noise corrupted' apply, but in the opposite direction. Assuming that a fuzzy set is totally characterized by a membership function, the membership function μ_{SMALL} (in the SMALL fuzzy set) is introduced to characterize the values associated with no movement and low-noise presence. By doing this, one can have a value between [0, 1] in order to measure the membership value with respect to the SMALL fuzzy set, where the value of 1 implies that the sample has no movement and low noise presence, and the value of 0 implies the opposite.

Thus, two fuzzy sets separately defined as BIG and SMALL are used to characterize the level of noise and/or movement in the sample processing. The membership functions μ_{BIG} and μ_{SMALL}, for gradients and angle deviations used by the temporal algorithm, are defined by the following expressions [25]:

$$\mu_{\text{SMALL}}(\chi) = \begin{cases} 1 & \text{if } \chi < \mu_1 \\ \exp\left\{ -\left((\chi - \mu_1)^2 / (2 \cdot \sigma^2) \right) \right\} & \text{otherwise} \end{cases}, \tag{8}$$

$$\mu_{\text{BIG}}(\chi) = \begin{cases} 1 & \text{if } \chi > \mu_2 \\ \exp\left\{ -\left((\chi - \mu_2)^2 / (2 \cdot \sigma^2) \right) \right\} & \text{otherwise} \end{cases}, \tag{9}$$

when χ = $\theta_{\beta\gamma}$ for angle deviations, one has to select the parameters, standard deviation σ = 0.3163, mean μ_1 = 0.2, and mean μ_2 = 0.615; when χ = $\nabla_{\beta\gamma}$ for gradient values, select the parameters, standard deviation σ = 31.63, mean μ_1 = 60, and mean μ_2 = 140. The parameter values were obtained through extensive simulations carried out on the color video sequences used in this study. The idea was to find the optimal parameter values according to the PSNR and MAE criteria. The procedure used to compute the optimal values of the parameters in the event that χ = $\theta_{\beta\gamma}$ is selected was the beginning and variation of standard deviation starting with the value 0.1, so the PSNR and MAE criteria could reach their optimal values while maintaining the fixed values of μ_1 = 0.1 and μ_2 = 0.1. Once we have the optimal values of PSNR and MAE, the parameter of standard deviation is fixed and μ_1 subsequently increases until it reaches the optimal values for the PSNR and MAE criteria. Finally, upon the fixing of the standard deviation and μ_1, the μ_2 is varied until it again reaches the optimal values for the PSNR and MAE criteria. The same approach is used to calculate the values of the parameters when the event χ = $\nabla_{\beta\gamma}$ is selected, based on the PSNR and MAE criteria. These experimental results were obtained using the well-known Miss America and Flowers color video sequences.

The fuzzy rules illustrated in Figure 5 are designed to detect, pixel by pixel, the presence of motion. First, the

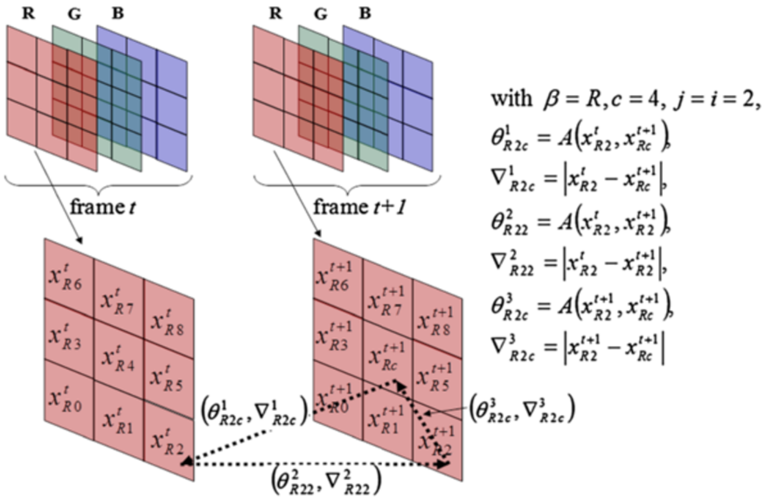
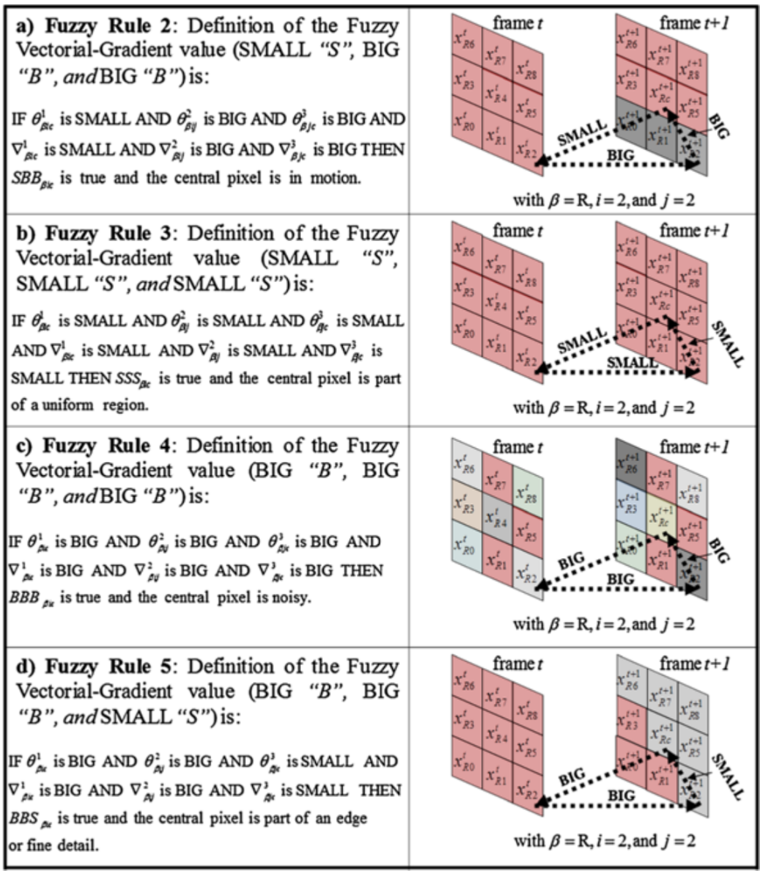

R
G
B
frame t
R
G
B
frame $t+1$
with $\beta = R, c = 4, \ j = i = 2,$
$\theta^{1}_{R2c} = A\left(x^{t}_{R2}, x^{t+1}_{Rc}\right),$
$\nabla^{1}_{R2c} = \left|x^{t}_{R2} - x^{t+1}_{Rc}\right|,$
$\theta^{2}_{R22} = A\left(x^{t}_{R2}, x^{t+1}_{R2}\right),$
$\nabla^{2}_{R22} = \left|x^{t}_{R2} - x^{t+1}_{R2}\right|,$
$\theta^{3}_{R2c} = A\left(x^{t+1}_{R2}, x^{t+1}_{Rc}\right),$
$\nabla^{3}_{R2c} = \left|x^{t+1}_{R2} - x^{t+1}_{Rc}\right|$
x^{t}_{R6}
x^{t}_{R7}
x^{t}_{R8}
x^{t}_{R3}
x^{t}_{R4}
x^{t}_{R5}
x^{t}_{R0}
x^{t}_{R1}
x^{t}_{R2}
x^{t+1}_{R6}
x^{t+1}_{R7}
x^{t+1}_{R8}
x^{t+1}_{R3}
x^{t+1}_{Rc}
x^{t+1}_{R5}
x^{t+1}_{R0}
x^{t+1}_{R1}
x^{t+1}_{R2}
$\left(\theta^{1}_{R2c}, \nabla^{1}_{R2c}\right)$
$\left(\theta^{3}_{R2c}, \nabla^{3}_{R2c}\right)$
$\left(\theta^{2}_{R22}, \nabla^{2}_{R22}\right)$

Figure 4 Application example for Equations 6 and 7, where $\beta = R$, and $i = j = 2$.



a) Fuzzy Rule 2: Definition of the Fuzzy Vectorial-Gradient value (SMALL "*S*", BIG "*B*", *and* BIG "*B*") is:

IF $\theta^{1}_{\beta ic}$ is SMALL AND $\theta^{2}_{\beta ij}$ is BIG AND $\theta^{3}_{\beta jc}$ is BIG AND $\nabla^{1}_{\beta ic}$ is SMALL AND $\nabla^{2}_{\beta ij}$ is BIG AND $\nabla^{3}_{\beta jc}$ is BIG THEN $SBB_{\beta ic}$ is true and the central pixel is in motion.

b) Fuzzy Rule 3: Definition of the Fuzzy Vectorial-Gradient value (SMALL "*S*", SMALL "*S*", *and* SMALL "*S*") is:

IF $\theta^{1}_{\beta ic}$ is SMALL AND $\theta^{2}_{\beta ij}$ is SMALL AND $\theta^{3}_{\beta jc}$ is SMALL AND $\nabla^{1}_{\beta ic}$ is SMALL AND $\nabla^{2}_{\beta ij}$ is SMALL AND $\nabla^{3}_{\beta jc}$ is SMALL THEN $SSS_{\beta ic}$ is true and the central pixel is part of a uniform region.

c) Fuzzy Rule 4: Definition of the Fuzzy Vectorial-Gradient value (BIG "*B*", BIG "*B*", *and* BIG "*B*") is:

IF $\theta^{1}_{\beta ic}$ is BIG AND $\theta^{2}_{\beta ij}$ is BIG AND $\theta^{3}_{\beta jc}$ is BIG AND $\nabla^{1}_{\beta ic}$ is BIG AND $\nabla^{2}_{\beta ij}$ is BIG AND $\nabla^{3}_{\beta jc}$ is BIG THEN $BBB_{\beta ic}$ is true and the central pixel is noisy.

d) Fuzzy Rule 5: Definition of the Fuzzy Vectorial-Gradient value (BIG "*B*", BIG "*B*", *and* SMALL "*S*") is:

IF $\theta^{1}_{\beta ic}$ is BIG AND $\theta^{2}_{\beta ij}$ is BIG AND $\theta^{3}_{\beta jc}$ is SMALL AND $\nabla^{1}_{\beta ic}$ is BIG AND $\nabla^{2}_{\beta ij}$ is BIG AND $\nabla^{3}_{\beta jc}$ is SMALL THEN $BBS_{\beta ic}$ is true and the central pixel is part of an edge or fine detail.

frame t
frame $t+1$
SMALL
BIG
BIG
with $\beta = \mathrm{R}, i = 2, \text{and } j = 2$
frame t
frame $t+1$
SMALL
SMALL
SMALL
with $\beta = \mathrm{R}, i = 2, \text{and } j = 2$
frame t
frame $t+1$
BIG
BIG
BIG
with $\beta = \mathrm{R}, i = 2, \text{and } j = 2$
frame t
frame $t+1$
BIG
BIG
SMALL
with $\beta = \mathrm{R}, i = 2, \text{and } j = 2$

Figure 5 Fuzzy rules 2, 3, 4, and 5 used to determine the motion level for *t* and *t* + 1 frames. (a) Fuzzy rule 2 $\mathrm{SBB}_{\beta ic}$. **(b)** Fuzzy rule 3 $\mathrm{SSS}_{\beta ic}$. **(c)** Fuzzy rule 4 $\mathrm{BBB}_{\beta ic}$. **(d)** Fuzzy rule 5 $\mathrm{BBS}_{\beta ic}$.

motion relative to the central pixel in the $t + 1$ frame is detected, using the pixels in the t frame; then, motion detection is performed on a pixel basis in both frames; and finally, this procedure applies only to the pixels of the $t + 1$ frame. Following this, the procedure for the proposed fuzzy rules is described in Figure 5; these fuzzy rules allow the analyst to characterize the presence of motion and/or noise in the sample in order to determine which procedure to utilize during the image processing.

The fuzzy rules of Figure 5 were designed to characterize, in a fuzzy directional manner, the relationship between pixels in a sliding window using two frames. Hence, the movement and the noise level presence in the central pixel of the sample are found. To understand the meaning of these fuzzy rules, the following situation is assumed: if the fuzzy directional values obtained by the membership function for the SMALL fuzzy set are close to one, then there is neither motion nor low-noise presence in the central pixel component. Conversely, if the values of the membership function are close to one for the BIG fuzzy set, the central pixel component is noisy and/or presents motion. Thus, for fuzzy rule 2, the values SMALL, BIG, and BIG (SBB) characterize a pixel in motion, in such a way that the first value characterizes the closeness of a SMALL component to the central pixel in the $t + 1$ frame with the pixel component of a neighbor in the t frame; the first BIG value indicates that the component of the pixel in the t frame and the component of the pixel in the $t + 1$ frame are unrelated; and the second BIG conveys that the value of the component of the pixel of the $t + 1$ frame, with respect to the component of the central pixel of the $t + 1$ frame shows some difference, therefore this pixel is highly likely to belong to an edge and/or is in motion. These findings reinforce the correctness of the parameters obtained for other neighboring component pixels. In this way, the relationship of proximity between the central pixel of the $t + 1$ frame with respect to the neighboring pixels of the t and $t + 1$ frames is obtained.

This study also aims at improving performance over computational resources of the algorithm making the distinction among different areas, especially, finding areas of an image that could be processed by a magnitude filter without affecting the fine image details and other image characteristics. The procedure to accomplish this is as follows: the sample standard deviation that includes the $3 \times 3 \times 2$ pre-processing window for each color channel in the t and $t + 1$ frames is calculated, thereby obtaining the parameter $\sigma_{\beta}^{''}$. This is described as the temporal SD because it is calculated over two frames (t and $t + 1$) of the video sequence. The procedure to calculate $\sigma_{\beta}^{''}$ is similar to that used in Section 2 but applied to a $3 \times 3 \times 2$ sample consisting of both frames. Then, it is compared with the SD $\sigma_{\beta}^{''}$ obtained for the spatial algorithm in Subsection 2.1, as follows: IF $\left\{\left(\sigma_{\text{red}}^{''} \geq 0.4\,\sigma_{\text{red}}^{'}\right) AND \left(\sigma_{\text{green}}^{''} \geq 0.4\,\sigma_{\text{green}}^{'}\right) AND \left(\sigma_{\text{blue}}^{''} \geq 0.4\,\sigma_{\text{blue}}^{'}\right)\right\}$, THEN fuzzy rules 2, 3, 4, and 5 are employed; otherwise, the *Mean Filter* is utilized. The AND operator therein is the 'logical AND'. Here, the value 0.4 in the condition sentence is selected to distinguish different areas containing fine details from those showing a uniform pattern. This value was found experimentally, according to the optimal PSNR and MAE values. Therefore, the application of the Mean Filter Algorithm implies that the uniform area is under processing:

$$\bar{y}_{\beta\text{out}} = \sum_{i=0}^{N} x_{\beta i}/N, \quad N = 17, \tag{10}$$

where $x_{\beta i}$ represents each one of the pixels in a $3 \times 3 \times 2$ pre-processing window, $N = 17$ is selected to take into account all pixel components in the two frames to be processed.

The general standard deviation used in this stage of the algorithm was adapted locally according to the pixels that agree with Figure 5 in the current sample. To acquire a new locally adapted SD, which will be used in the next frame of video sequence, a sensitive parameter α must be introduced describing the current distribution of the pixels and featuring a measure of temporal relationship between the t and $t + 1$ frames. The main idea of the sensitive parameter is to control the amount of filtering; this parameter modifies its value on its own to agree with the locally adapted SD. The same parameter allows the upgrading of the SD that helps to describe the relationship in the frames t and $t + 1$, producing a temporal parameter. When the Mean Filter is applied, the sensitivity parameter value is $\alpha = 0.125$.

In case there is a drastic change in the fine details, edges, and movements in the current samples, these will be reflected in their parameter values - such as the membership functions, the SD, and the sensitivity parameters, as well as in their fuzzy vectorial-gradient values. The consequences, which are applied for each fuzzy rule, are based on the different conditions present in the sample.

The updating of the general standard deviation that should be used in the processing of the next frame is performed according to the expression:

$$\sigma_{\beta}^{'} = \left(\alpha \cdot \left({}^{\sigma_{\text{total}}}/_{5}\right)\right) + (1-\alpha) \cdot \left(\sigma_{\beta}^{'}\right). \tag{11}$$

The aim of this equation is to control the locally adapted spatial SD and, in the same manner, control the temporal SD which will, on its turn, control the amount of filtering modifying the T_β threshold as will be shown later.

Parameters $\sigma'_\beta, \sigma''_\beta$, and σ_{total} describe how the pixels in the t and $t + 1$ frames are related to each other in a spatial $\left(\sigma'_\beta\right)$ and temporal $\left(\sigma''_\beta, \text{and } \sigma_{\text{total}}\right)$ way. The SD updating of σ_{total} is achieved through: $\sigma_{\text{total}} = \left(\sigma''_{\text{Red}}+\sigma''_{\text{Green}}+\sigma''_{\text{Blue}}\right)\Big/3$; this is the average value of the temporal SD using the three color planes of the images. This relationship is designed to have the other color components of the image contribute to the sensitivity parameter.

The structure of Equation 11 can be illustrated using an example: if the Mean Filter Algorithm was selected for application instead of fuzzy rules 2, 3, 4, and 5, the sensitive parameter $\alpha = 0.125$ used for the algorithm describes that the t and $t + 1$ frames are closely related. This means that the pixels in the t frame bear low noise due to the fact that the spatial algorithm was applied to this frame (see Subsection 2.1) and that the pixels in the $t + 1$ frame are probably low-noise too. However, at this time, because the t frame has only been filtered by the spatial algorithm (see Subsection 2.1), it seems better to increase the weight obtained by the t frame in the spatial SD $\left(\sigma'_\beta\right)$, rather than using that obtained by the $t + 1$ frame $\left(\text{temporal SD}\left(\sigma''_\beta\right)\right)$. That is why the weights of σ_{total} multiplied by $\alpha = 0.125$, and the weight of σ'_β multiplied by $(1 - \alpha) = 0.875$ are used.

The application of fuzzy rules to pixels allows a better preservation of the inherent characteristics of the color images. The following methodology is based on these concepts, using the pixels indicated by each fuzzy rule in the process of noise suppression. That is: if the number of pixels presented in the next condition, (1) IF {(# $\text{pixels}_{\text{SBB}}$ > # $\text{pixels}_{\text{SSS}}$)AND(# $\text{pixels}_{\text{SBB}}$ > # $\text{pixels}_{\text{BBB}}$) AND(# $\text{pixels}_{\text{SBB}}$ > # $\text{pixels}_{\text{BBS}}$)}, is the biggest as compared to the other ones in the following IF conditions: (2) IF {(# $\text{pixels}_{\text{SSS}}$ > # $\text{pixels}_{\text{SBB}}$)AND(# $\text{pixels}_{\text{SSS}}$ > # $\text{pixels}_{\text{BBB}}$) AND(# $\text{pixels}_{\text{SSS}}$ > # $\text{pixels}_{\text{BBS}}$)}, (3) IF {(# $\text{pixels}_{\text{BBS}}$ > # $\text{pixels}_{\text{SBB}}$)AND(# $\text{pixels}_{\text{BBS}}$ > # $\text{pixels}_{\text{SSS}}$)AND(# $\text{pixels}_{\text{BBS}}$ > # $\text{pixels}_{\text{BBB}}$)}, and (4) IF {(# $\text{pixels}_{\text{BBB}}$ > # $\text{pixels}_{\text{SBB}}$)AND (# $\text{pixels}_{\text{BBB}}$ > # $\text{pixels}_{\text{SSS}}$)AND(# $\text{pixels}_{\text{BBB}}$ > # $\text{pixels}_{\text{BBS}}$)}, the following methodology is applied to only those pixels that fulfill the condition:

$$y_{\beta\text{out}} = \sum_{i=1}^{\#\text{pixels}} x^{t-1}_{\beta i} \cdot \text{SBB}_{\beta i} \Bigg/ \sum_{i=1}^{\#\text{pixels}} \text{SBB}_{\beta i}, \tag{12}$$

where $x^t_{\beta i}$, and $x^{t+1}_{\beta i}$ represent each pixel in the t and $t + 1$ frames that fulfills the assumed fuzzy rule conditions, respectively, with $\alpha = 0.875$. For a better understanding of the use of fuzzy rules, see Figure 6. The following equations are used in cases where the largest number of pixels compared to the others is, for example, in case of the second condition (if #$\text{pixels}_{\text{SSS}}$ is the biggest, that means: # $\text{pixels}_{\text{SSS}}$ > # $\text{pixels}_{\text{SBB}}$ > # $\text{pixels}_{\text{BBB}}$ > # $\text{pixels}_{\text{BBS}}$) we perform:

$$y_{\beta\text{out}} = \sum_{i=1}^{\#\text{pixels}} \left(x^{t-1}_{\beta i} \cdot 0.5 + x^{t}_{\beta i} \cdot 0.5\right) \cdot \text{SSS}_{\beta i} \Bigg/ \sum_{i=1}^{\#\text{pixels}} \text{SSS}_{\beta i}, \tag{13}$$

where $\alpha = 0.125$; or for the third condition (# $\text{pixels}_{\text{BBS}}$ is the biggest):

$$y_{\beta\text{out}} = \sum_{i=1}^{\#\text{pixels}} x^{t}_{\beta i} \cdot \left(1\text{–}\text{BBS}_{\beta i}\right) \Bigg/ \sum_{i=1}^{\#\text{pixels}} \left(1\text{–}\text{BBS}_{\beta i}\right) \tag{14}$$

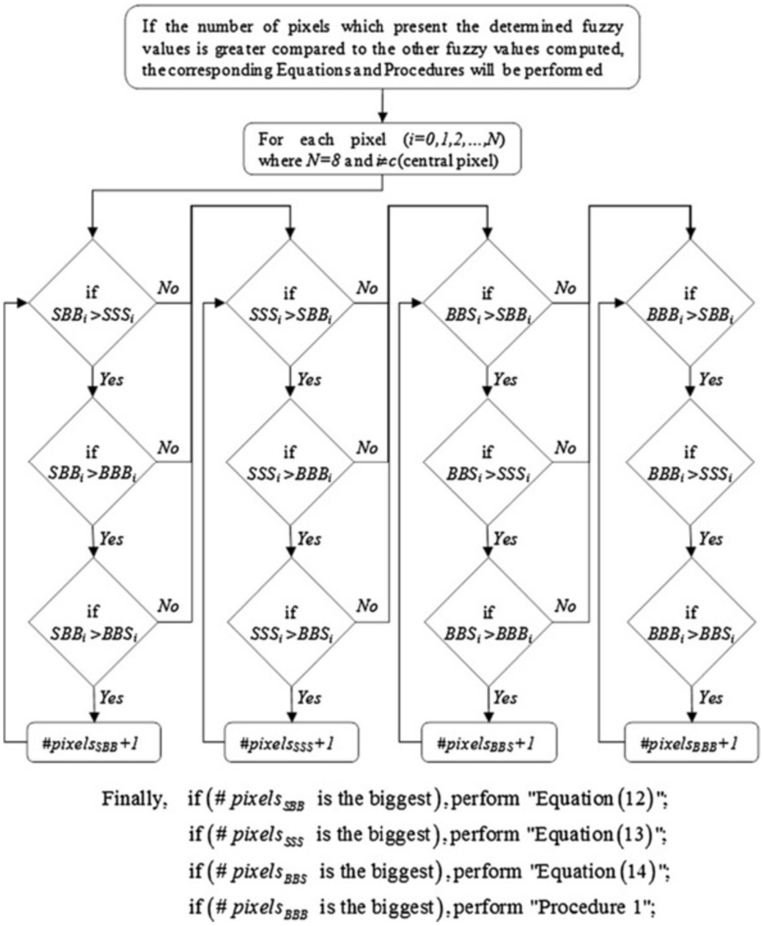
where $\alpha = 0.875$. Finally, for the fourth condition (# $\text{pixels}_{\text{BBB}}$ is the biggest), when the number of pixels (# pixels) with $\text{BBB}_{\beta i}$ value is the biggest, the next algorithm is performed:

Procedure 1: consider the nine fuzzy vectorial-gradient values obtained from the $\text{BBB}_{\beta i}$ values. The central value is selected along with the three neighboring fuzzy values in order to detect the motion. The conjunction of the four subfacts is performed, which are combined by a triangular norm [19]. The intersection of all possible combinations of $\text{BBB}_{\beta i}$ and three different neighboring membership degrees gives 56 values to be obtained: $C^K_{N-1} = \binom{N-1}{K} = 56$, where $N = 9$, and with $K = 3$ elements are to be included in the intersection process. The values are added using an algebraic equation (sum = $A + B - A \cdot B$) [19] of all instances in order to obtain the *motion-noise* confidence parameter.

The motion-noise confidence parameter is used to update the SD and to obtain the output pixel by means of the next algorithm: $y_{\beta\text{out}} = (1\text{–}\alpha)\cdot x^{t+1}_{\beta c} + \alpha\cdot x^{t}_{\beta c}$, (where $\alpha = 0.875$ if the motion-noise = 1; and $\alpha = 0.125$ when the motion-noise = 0). If there is no majority in the number of pixels to any of the fuzzy rules, then the output pixel is computed as follows: $y_{\beta\text{out}} = 0.5\cdot x^{t+1}_{\beta c} + 0.5\cdot x^{t}_{\beta c}$, where $\alpha = 0.5$.

Finally, the algorithm employs the above-outlined spatial algorithm for smoothing the non-stationary noise remaining after application of the temporal filter, with the only modification in its threshold value of $T_\beta = 0.25\sigma'_\beta$, in agreement with Figure 1.

In summary, all parameters and their optimal values used in the development of this algorithm is given in the Table 1. All the optimum parameters were found under numerous simulations using different video color sequences with different levels of Gaussian noise and with different criteria to characterize noise suppression (PSNR), details and edges preservation (MAE), and



Finally, if (# $pixels_{SBB}$ is the biggest), perform "Equation (12)";
if (# $pixels_{SSS}$ is the biggest), perform "Equation (13)";
if (# $pixels_{BBS}$ is the biggest), perform "Equation (14)";
if (# $pixels_{BBB}$ is the biggest), perform "Procedure 1";

Figure 6 The denoising scheme applied in the temporal algorithm. In the case of movement, uniform region, noise, edge, and fine detail, agrees with Figure 5.

chromaticity preservation (normalized chromaticity deviation (NCD)).

All the other parameters used in the algorithm are locally updated in agreement with the adaptive method; this means that these parameters change locally in all the sequences of the video frames.

3. Simulation results

The results presented show the effectiveness of the proposed algorithm against others used for comparison. To accomplish this, video sequences containing different features and textures were used: Miss America, Flowers, and Chair sequences; all of them contaminated by Gaussian noise with a variance (VAR) 0.0 to 0.05. The color video sequences processed for this work were 24-bit true color and 176 × 144 pixels (QCIF format).

Table 1 All parameters used in the algorithm and their optimal values

Parameters with optimal values used in the algorithm			
$\tau_1 = 0.1$ (threshold used in the 'Spatial Filtering Algorithm' used only once)			
$\tau_\beta = 0.25\sigma_\beta^{'}$ (threshold value used in the whole video sequence)			
when $\chi = \theta_{\beta\gamma}$	$\sigma = 0.3163$	$\mu_1 = 0.2$	$\mu_2 = 0.615$
when $\chi = \nabla_{\beta\gamma}$	$\sigma = 31.63$	$\mu_1 = 60$	$\mu_2 = 140$
$\alpha = 0.125$ (sensitivity parameter)			
$1 - \alpha = 0.875$ (complement of the sensitive parameter)			

Figure 7 shows the frames of the original video sequences subjectively used to characterize the noise suppression, detail and edge preservation, and the chromaticity. The filtered frames and complete video sequences were quantitatively evaluated according to the following criteria: PSNR was used to characterize the noise suppression capabilities (a larger PSNR reflects a better preservation of the characteristics of video frames); the MAE was used to numerically measure the level of preservation of edges and fine details; and the NCD was used to characterize the perceptual error between two color vectors, according to the human perception of color [22,26]. These criteria were applied to the proposed framework and compared with several algorithms that have demonstrated beneficial properties in video denoising.

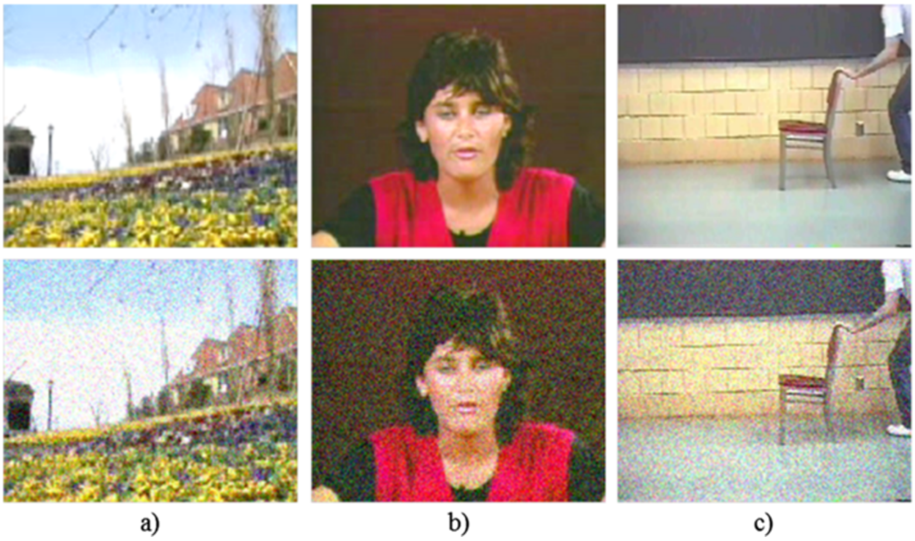


Figure 7 Original and corrupted images used to subjectively evaluate proposed and comparative algorithms. (a) 10th Flowers video sequence frame, **(b)** 10th Miss America video sequence frame, and **(c)** 10th Chair video sequence frame. Frames are corrupted with VAR = 0.01.

The proposed 'Fuzzy Directional Adaptive Recursive Temporal Filter for Gaussian Denoising' algorithm, referred to as FDARTF_G, was compared with others, the FLRSTF algorithm that uses similar fuzzy techniques [19], the FLRSTF_ANGLE, the VGVDF_G, and the VMMKNN_G [21,22] algorithm that uses order statistics techniques for the removal of Gaussian noise.

Figure 8 illustrates the denoising capability and preservation ability of all mentioned filters for the 10th frame of the Miss America and Flowers video sequences. This figure shows that the designed framework produces the best results. The criteria applied are the PSNR and MAE. Here, one can observe that the performance of our design is the best for the Miss America frame; on the other hand, for the Flowers frame, the best results are generated by the PSNR criterion for the majority of the noise levels, while for the MAE criterion, the best results are for low-noise levels.

The processing results in the cases of the 20th and 30th frames for the three video sequences with corruption levels of VAR = 0.005 and VAR =0.01 have shown that the best performances in the MAE, PSNR, and NCD criteria are most of the times achieved through applying the proposed algorithm, as shown in Table 2.

A more sophisticated filter used as a comparison is the CBM3D [17]; this filter works in other domain, which

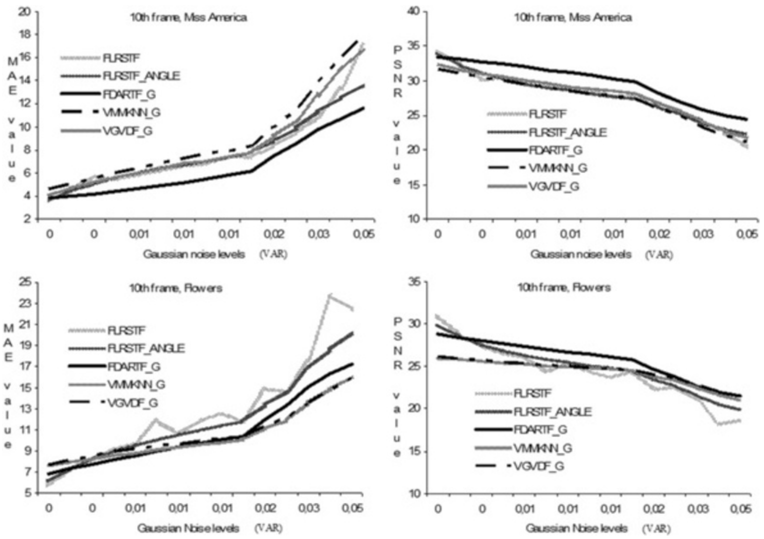


Figure 8 MAE and PSNR criteria for the 10th Miss America and Flowers frames.

Table 2 Comparative restoration results agree with the MAE, PSNR, and NCD criteria

Algorithm	Criteria	Gaussian noise VAR = 0.005						Gaussian noise VAR = 0.01					
		Chair		Miss America		Flowers		Chair		Miss America		Flowers	
		Frame 20	Frame 30	Frame 20	Frame 30	Frame 20	Frame 30	Frame 20	Frame 30	Frame 20	Frame 30	Frame 20	Frame 30
FLRSTF	MAE	6.375	6.613	5.818	5.622	9.628	9.551	8.400	8.325	7.477	7.362	11.932	11.652
	PSNR	28.979	28.720	29.926	30.157	26.192	26.263	26.701	26.589	27.686	27.720	24.363	24.559
	NCD	0.0110	0.0112	0.0263	0.0261	0.0167	0.0168	0.0140	0.0137	0.0257	0.0260	0.0205	0.0203
FLRSTF_ANGLE	MAE	6.374	6.620	5.826	5.629	9.825	9.643	8.458	8.319	7.500	7.386	11.971	11.661
	PSNR	28.968	28.716	29.905	30.133	26.007	26.164	26.638	26.605	27.681	27.687	24.340	24.556
	NCD	0.0110	0.0112	0.0263	0.0262	0.0175	0.0171	0.0140	0.0137	0.0258	0.0259	0.0205	0.0203
FDARTF_G	MAE	5.501	5.603	4.459	4.279	8.503	8.281	7.416	7.245	6.069	5.909	10.438	10.056
	PSNR	29.170	29.291	32.510	32.917	27.309	27.611	27.454	27.391	30.059	30.300	25.717	26.054
	NCD	0.0105	0.0106	0.0219	0.0216	0.0155	0.0151	0.0135	0.0128	0.0220	0.0216	0.0185	0.0182
VMMKNN_G	MAE	7.987	7.886	6.178	6.103	8.777	8.459	9.250	9.677	8.143	8.081	9.916	9.634
	PSNR	25.368	25.607	29.799	29.826	25.348	25.801	25.159	24.427	27.612	27.683	24.629	24.978
	NCD	0.0146	0.0141	0.0286	0.0285	*0.0153*	*0.0147*	0.0174	0.0164	0.0287	0.0282	0.0167	0.0167
CBM3D	MAE	*4.863*	*4.854*	*3.512*	*3.509*	*7.821*	*7.795*	*7.712*	*7.729*	*4.518*	*4.526*	*9.934*	*9.896*
	PSNR	*33.185*	*33.193*	*33.658*	*33.931*	*30.790*	*30.670*	*32.1*	*32.06*	*32.58*	*32.87*	*29.45*	*29.363*
	NCD	-	-	-	-	-	-	-	-	-	-	-	-

For the 20th and 30th frames of the video sequences used. Numbers in italics indicate best results which agreed with the method and criteria implemented.

consists of two steps in which blocks are grouped by spatio-temporal predictive blockmatching and each 3D group is filtered by a 3D transform domain shrinkage, and the complex *3D wavelet transform method 3DWF* shows better results in terms of PSNR and MAE criteria than our proposed filter. For the Flowers sequence, the received results for our algorithm 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. Advantages to take into account in our filtering method are the prevention/avoidance of spatiotemporal blur; one should only consider neighboring pixels from the current frame in case of detected motion. Other advantage is in preserving the details in the frame content; the filtering should not be as 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, where the eye is not sensitive enough to detect this. In the case of homogeneous areas, strong filtering should be performed to remove as much noise as possible. The performance of our methodology is similar to the achieved in the paper of Mélange et al. [27], and it was outperformed by CBM3D method too.

Table 3 presents average results for all of the proposed criteria from all of the frames that form the video sequences used. Based on these results, one can state that the best performance response is by the proposed filtering algorithm (FDARTF_G) for all the Gaussian noise levels for the Miss America video sequence. In the Flowers video sequence, the best results are achieved by the PSNR criterion for the majority of the noise levels. Additionally, the use of the MAE and NCD criteria achieves very good results in the preservation of details and chromatic properties.

In Figure 9, we see that for the Chair video sequence, the best performance is given by our proposed method for every frame forming the video at medium and high noise levels. The best results were obtained for all of the criteria used (PSNR, MAE, and NCD). Evidently, the CVBM3D version, to process video color images, [17] filtering method will deliver better results against our

Table 3 The *PSNR*, *MAE*, and *NCD* criteria averaged results

Gaussian noise (VAR)			0.001	0.005	0.01	0.015	0.02	0.03	0.04	0.05
Miss America	MAE	FLRSTF	3.624	5.967	7.625	8.853	9.847	11.414	12.999	14.477
		FDARTF_G	*3.549*	*4.542*	*6.126*	*7.453*	*8.465*	*9.832*	*10.822*	*11.694*
		VMMKNN_G	4.377	6.217	8.198	9.871	11.372	13.934	16.072	17.921
		VGVDF_G	3.71	5.685	7.419	8.92	10.253	12.563	14.609	16.441
	PSNR	FLRSTF	34.303	29.73	27.573	26.267	25.328	23.998	22.827	21.888
		FDARTF_G	*34.013*	*32.303*	*29.929*	*28.258*	*27.127*	*25.811*	*25.02*	*24.404*
		VMMKNN_G	32.057	29.689	27.55	26.045	24.887	23.23	22.083	21.213
		VGVDF_G	33.048	30.384	28.383	26.921	25.789	24.111	22.847	21.85
	NCD	FLRSTF	0.013	0.021	0.027	0.031	0.034	0.039	0.044	0.049
		FDARTF_G	*0.013*	*0.016*	*0.022*	*0.027*	*0.031*	*0.036*	*0.04*	*0.042*
		VMMKNN_G	0.015	0.022	0.029	0.034	0.04	0.049	0.058	0.066
		VGVDF_G	0.013	0.021	0.027	0.033	0.037	0.046	0.053	0.06
Flowers	MAE	FLRSTF	*6.011*	9.866	12.013	13.823	14.911	17.142	20.408	20.664
		FDARTF_G	7.295	*8.847*	10.647	12.262	13.539	15.288	16.464	17.39
		VMMKNN_G	7.754	9.07	*10.31*	*11.357*	*12.258*	*13.896*	15.346	16.642
		VGVDF_G	8.04	9.588	10.786	11.736	12.571	14.036	*15.331*	*16.459*
	PSNR	FLRSTF	*30.289*	25.969	24.289	23.052	22.411	21.182	19.569	19.525
		FDARTF_G	28.139	*26.906*	*25.427*	*24.255*	*23.424*	*22.375*	*21.722*	*21.24*
		VMMKNN_G	26.128	25.324	24.523	23.858	23.303	22.347	21.564	20.907
		VGVDF_G	25.765	24.835	24.135	23.602	23.139	22.358	21.693	21.127
	NCD	FLRSTF	*0.011*	0.018	0.021	0.025	0.026	0.03	0.033	0.036
		FDARTF_G	0.014	0.017	0.02	0.023	0.025	0.028	0.03	0.032
		VMMKNN_G	0.014	*0.016*	*0.018*	*0.02*	*0.021*	*0.023*	*0.025*	*0.027*
		VGVDF_G	0.017	0.019	0.021	0.023	0.024	0.026	0.028	0.03

For the frames of the *Miss America* and *Flowers* video sequences corrupted by different Gaussian Noise levels. Numbers in italics indicate best results which agreed with the method and criteria implemented.

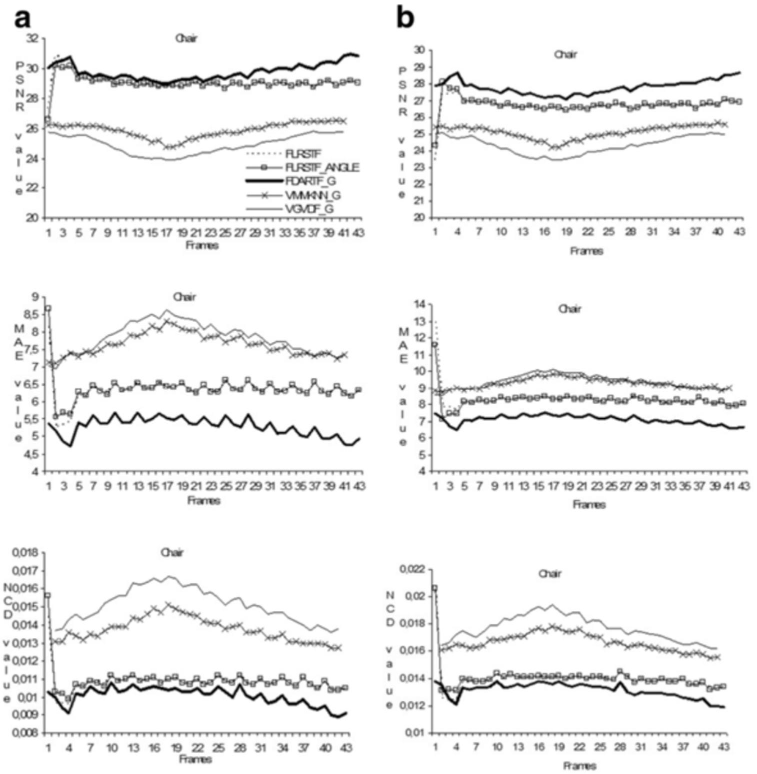


Figure 9 PSNR, MAE, and NCD criteria for the Chair video sequence. With column **(a)** VAR = 0.005 and column **(b)** VAR = 0.01.

suggestion, since 5 dB through 8 dB in the PSNR criterion, because it is more sophisticated and works in different domain and uses complex 3D wavelet transform method 3DWF that makes it powerful even though, until now, algorithms are not more powerful as those suggested in [17], as the methods proposed by Yu et al. [28], by Priyam Chatterjee and Milanfar [29], by Zuo et al. [30], and by Li et al. [31].

Finally, in Figure 10, one can see the filtered frames after different algorithms were used to estimate the quality of the subjective vision perception. From the results of the proposed algorithm, it is easy to corroborate that

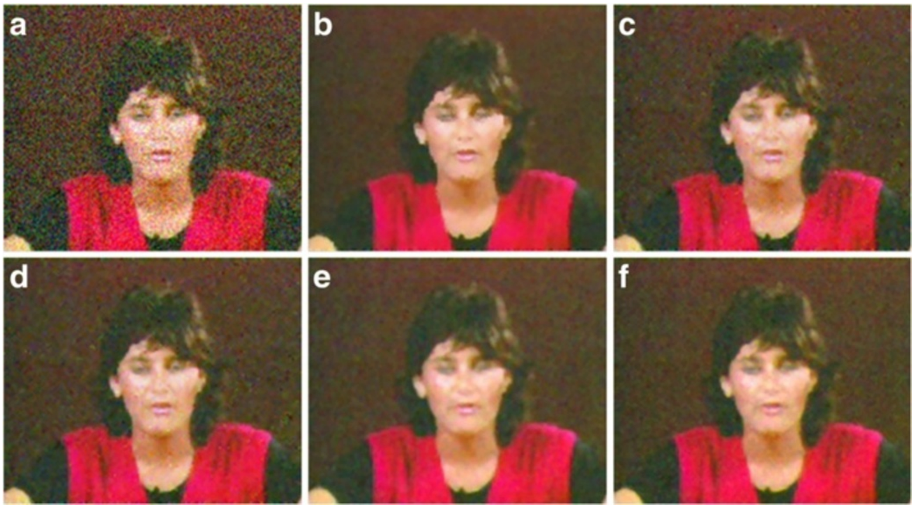


Figure 10 The filtered 10th frames of the Miss America video sequence. (a) Frame corrupted by Gaussian noise, VAR = 0.005, **(b)** FDARTF_G, **(c)** FLRSTF, **(d)** FLRSTF_ANGLE, **(e)** VGVDF_G, and **(f)** VMMKNN_G.

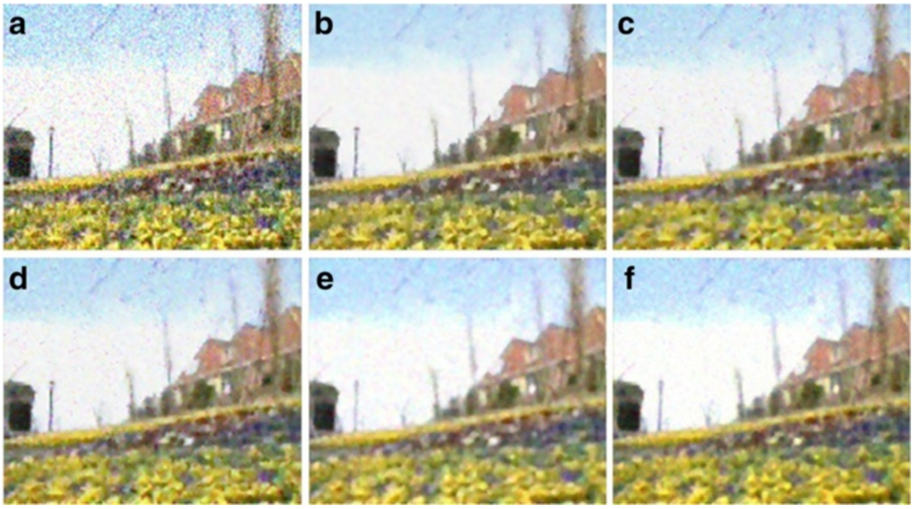


Figure 11 The filtered 10th frames of the Flowers video sequence. (a) Image corrupted by Gaussian noise, VAR = 0.01, **(b)** FDARTF_G, **(c)** FLRSTF, **(d)** FLRSTF_ANGLE, **(e)** VGVDF_G, and **(f)** VMMKNN_G.

this filter has the best performance in detail preservation and noise suppression. In the FDARTF_G filtered image, one can observe cleaner regions with better preservation of fine details and edges, as compared to other algorithms.

In Figure 11 below, the proposed framework produces the best results in the areas of detail preservation and noise suppression. One can perceive (in the vicinity of the tree) that in the case of the FDARTF_G filtering, the resulting image is less influenced by noise compared to the image produced by other filters. In addition, the new filter preserves more details of the features displayed in the background environment.

From Figure 12, one can see that the proposed framework achieves the best results in details, edges, and preservation of chromaticity. We can observe that the uniform regions are free from noise influence in the case of the FDARTF_G filtering than with the other filters implemented. Also, the new filter preserves more details in the features seen in the background environment.

Since the proposed algorithm is adaptive, it is difficult to obtain computational information related to how many

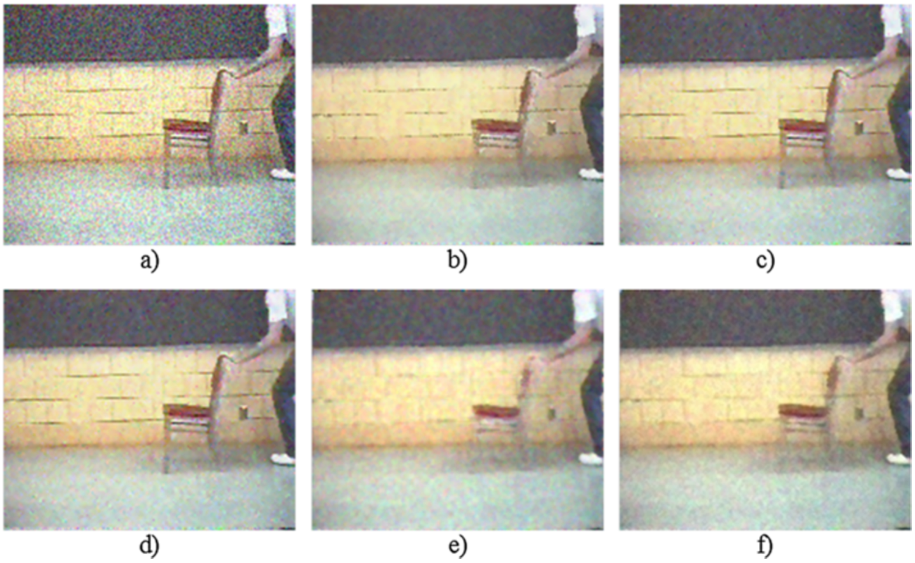


Figure 12 The filtered 10th frames of the Chair video sequence. (a) Image corrupted by Gaussian noise, VAR = 0.01, **(b)** FDARTF_G, **(c)** FLRSTF, **(d)** FLRSTF_ANGLE, **(e)** VGVDF_G, and **(f)** VMMKNN_G.

adds, multiplies, or divisions among other operations like trigonometrical ones were carried out; we provide real-time performance using a DSP from Texas Instruments, Dallas, TX, USA; this was the DM642 [32] giving the following results: for our proposed FDARTF_G, it spent an average time of 17.78 s per frame, but in a complete directional (VGVDF_G) processing algorithm, it spent an average time of 25.6 s per frame, both in a QCIF format.

4. Conclusions

The fuzzy and directional techniques working together have proven to be a powerful framework for image filtering applied in color video denoising in QCIF sequences. This robust algorithm performs motion detection and local noise standard deviation estimation. These proper video-sequence characteristics have been obtained and converted into parameters to be used as thresholds in different stages of the novel proposed filter. This algorithm permits the processing of t and $t + 1$ video frames, producing an appreciable savings of time and resources expended in computational filtering.

Using the advantages of both techniques (directional and diffuse), it was possible to design an algorithm that can preserve edges and fine details of video frames besides maintaining their inherent color, improving the preservation of the texture of the colors versus results obtained by the comparative algorithms. Other important conclusion is that for sequences obtained by a still camera, our method has a better performance in terms of PSNR than other multiresolution filters of a *similar complexity*, but it is outperformed by some more sophisticated methods (CBM3D).

The simulation results under the proposed criteria PSNR, MAE, and NCD were used to characterize an algorithm's efficiency in noise suppression, fine details, edges, and chromatic properties preservation. The perceptual errors have demonstrated the advantages of the proposed filtering approach.

Competing interests

The authors declare that they have no competing interest.

Acknowledgements

The authors thank the Instituto Politécnico Nacional de México (National Polytechnic Institute of Mexico) and CONACYT for their financial support.

Received: 2 March 2012 Accepted: 12 March 2014
Published: 2 April 2014

References

1. AJ Rosales-Silva, FJ Gallegos-Funes, V Ponomaryov, Fuzzy Directional (FD) Filter for impulsive noise reduction in colour video sequences. J. Vis. Commun. Image Represent. **23**(1), 143–149 (2012)
2. A Amer, H Schrerder, A new video noise reduction algorithm using spatial subbands. Int. Conf. on Electronic Circuits and Systems **1**, 45–48 (13-16 October 1996)
3. G De Haan, IC for motion-compensated deinterlacing, noise reduction, and picture rate conversion. IEEE Trans. On Consumers Electronics **45**(3), 617–624 (1999)
4. R Rajagopalan, M Orchard, Synthesizing processed video by filtering temporal relationships. IEEE Trans. Image Process. **11**(1), 26–36 (2002)
5. V Seran, LP Kondi, New temporal filtering scheme to reduce delay in wavelet-based video coding. IEEE Trans. Image Process. **16**(12), 2927–2935 (2007)
6. V Zlokolica, M De Geyter, S Schulte, A Pizurica, W Philips, E Kerre, Fuzzy logic recursive change detection for tracking and denoising of video sequences, in *Paper presented at the IS&T/SPIE Symposium on Electronic Imaging, San Jose, California, USA, 14 March 2005*. doi: 10.1117/12.585854
7. A Pizurica, V Zlokolica, W Philips, Noise reduction in video sequences using wavelet-domain and temporal filtering, in *Paper presented at the SPIE Conference on Wavelet Applications in Industrial Processing, USA, 27 February 2004*. doi:10.1117/12.516069
8. W Selesnick, K Li, Video denoising using 2d and 3d dual-tree complex wavelet transforms, in *Paper presented at the Proc. SPIE on Wavelet Applications in Signal and Image Processing, USA, volume 5207, pp. 607-618; 14 November 2003*. doi: 10.1117/12.504896
9. N Rajpoot, Z Yao, R Wilson, Adaptive wavelet restoration of noisy video sequences, in *Paper presented at the IEEE International Conference on Image Processing, pp. 957-960, October 2004*. doi: 10.1109/ICIP.2004.1419459
10. C Ercole, A Foi, V Katkovnik, K Egiazarian, *Spatio-temporal pointwise adaptive denoising of video: 3d nonparametric regression approach* (Paper presented at the First Workshop on Video Processing and Quality Metrics for Consumer Electronics, January, 2005)
11. D Rusanovskyy, K Egiazarian, *Video denoising algorithm in sliding 3D DCT domain. Lecture Notes in Computer Science 3708* (Springer Verlag, Advanced Concepts for Intelligent Vision Systems, 2005), pp. 618–625
12. V Ponomaryov, A Rosales-Silva, F Gallegos-Funes, *Paper presented at the Proc. of SPIE-IS&T, Published in SPIE Proceedings Vol. 6811: Real-Time Image Processing 2008*, 4 March 2008. doi:10.1117/12.758659
13. G Varghese, Z Wang, Video denoising based on a spatio-temporal Gaussian scale mixture model. IEEE Trans. Circ. Syst. Video. Tech. **20**(7), 1032–1040 (2010)
14. L Jovanov, A Pizurica, S Schulte, P Schelkens, A Munteanu, E Kerre, W Philips, Combined wavelet-domain and motion-compensated video denoising based on video codec motion estimation methods. IEEE Trans. Circ. Syst. Video. Tech. **19**(3), 417–421 (2009)
15. J Dai, C Oscar, W Yang, C Pang, F Zou, X Wen, Color video denoising based on adaptive color space conversion, in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), June 2010, pp. 2992-2995*. doi: 10.1109/ISCAS.2010.5538013
16. C Liu, WT Freeman, A high-quality video denoising algorithm based on reliable motion estimation, in *Paper presented at the Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III* (Springer-Verlag, Heraklion, Crete, Greece, 2010), pp. 706–719
17. K Dabov, A Foi, K Egiazarian, Video denoising by sparse 3D transform-domain collaborative filtering, in *Proc. 15th European Signal Processing Conference, EUSIPCO 2007* (, Poznan, Poland, September 2007)
18. J Mairal, G Sapiro, M Elad, Learning multiscale sparse representations for image and video restoration. SIAM Multiscale Modeling and Simulation **7**(1), 214–241 (2008)
19. V Zlokolica, S Schulte, A Pizurica, W Philips, E Kerre, Fuzzy logic recursive motion detection and denoising of video sequences. J. Electron. Imag. **15**(2), 1–13 (2006). doi:10.1117/1.2201548
20. PE Trahanias, D Karakos, AN Venetsanopoulos, Directional processing of color images: theory and experimental results. IEEE Trans. Image Process. **5**(6), 868–880 (1996)
21. VI Ponomaryov, Real-time 2D-3D filtering using order statistics based algorithms. J. Real-Time Image Proc. **1**(3), 173–194 (2007)
22. V Ponomaryov, A Rosales-Silva, V Golikov, Adaptive and vector directional processing applied to video color images. Electron. Lett. **42**(11), 1–2 (2006)
23. Arizona State University, http://trace.eas.asu.edu/yuv/, October-2010
24. J Zheng, KP Valavanis, JM Gauch, Noise removal from color images. J. Intell. Robot. Syst. **7**, 3 (1993)
25. KN Plataniotis, AN Venetsanopoulos, *Color Image Processing and Applications* (Springer-Verlag, 26 May 2000)
26. A Pearson, Fuzzy Logic Fundamentals. Chapter 3, 2001, pp. 61–103. www.informit.com/content/images/0135705991/samplechapter/0135705991.pdf. August 2008
27. T Mélange, M Nachtegael, EE Kerre, V Zlokolica, S Schulte, VD Witte, A Pizurica, W Philips, Video denoising by fuzzy motion and detail adaptive

averaging. J. Electron. Imag. **17**(4), 043005-1–043005-19 (2008). http://dx.doi.org/10.1117/1.2992065

28. S Yu, O Ahmad, MNS Swamy, Video denoising using motion compensated 3-D wavelet transform with integrated recursive temporal filtering. IEEE Trans. Circ. Syst. Video. Tech. **20**(6), 780–791 (2010)
29. P Chatterjee, P Milanfar, Clustering-based denoising with locally learned dictionaries. IEEE Trans. Image Process. **18**(7), 1438–1451 (2009)
30. C Zuo, Y Liu, X Tan, W Wang, M Zhang, Video denoising based on a spatiotemporal Kalman-bilateral mixture model. Scientific World Journal (Hindawi) (2013)
31. S Li, H Yin, L Fang, Group-sparse representation with dictionary learning for medial image denoising and fusion. IEEE Transaction on Biomedical Engineering **59**(12) (2012)
32. Texas Instruments. http://www.ti.com/tool/tmdsevm642, January 2008

doi:10.1186/1687-5281-2014-20
Cite this article as: Rosales-Silva *et al.*: **Robust fuzzy scheme for Gaussian denoising of 3D color video.** *EURASIP Journal on Image and Video Processing* 2014 **2014**:20.