FUZZY VECTORIAL DIRECTIONAL PROCESSING IN DENOISING OF MULTICHANNEL IMAGES

Volodymyr Ponomaryov[1], Alberto Rosales[1], Francisco Gallegos[2], Antonio Romero[1]

[1]National Polytechnic Institute of Mexico, ESIME – Culhuacan

Av. Santa Ana, 1000, Col. San. Fco. Culhuacan, 04430, Mexico-city, Mexico

[2]National Polytechnic Institute of Mexico, ESIME-Zacatenco

vponomar@ipn.mx (*V.Ponomaryov*)

Abstract

We propose a fuzzy logic recursive scheme using gradients and vectors (directional processing) for motion detection and spatial-temporal filtering to decrease a Gaussian noise corruption. The usage of the spatial-temporal information is considered to be more efficient compared with the use of only temporal information in the presence of fast motion and low noise that is common for different algorithms that can be found in literature. We introduce novel ideas that employ the differences between images. That permits to connect images using angle deviations, obtaining several parameters and applying them in the robust algorithm that is capable to detect and differentiate movement on background of noise in any way. The proposed method takes into account several characteristics inherent in the images. We can separate them and improve processing time consuming, having only the robust process with those samples that demonstrate that their corruption and movement are in a high level. It has been demonstrated that taking into account, both characteristics (gradients and vectors) and connecting them together, we can realize the algorithm with better performance, improving the techniques that use such characteristics in a separate form. During the simulations it has been investigated two different video sequences to qualify effectiveness of this filter. Both sequences: "Miss America" and "Flowers" present different image texture to provide a better understanding in the robustness of the fuzzy logic algorithm and have shown the effectiveness of proposed fuzzy logic algorithms.

Keywords: Fuzzy Logic, Video Sequences, Motion, Vectors.

Presenting Author's biography

Volodymyr I. Ponomaryov: Ph.D. (1974), Doctor of Sciences (1981), Full Professor (1984). He is working as Professor at National Polytechnic Institute of Mexico (Mexico-city). Main research activity: signal/image processing, real time filtering, medical sensors, remote sensing, etc. He is author more than 100 scientific papers, 250 conference papers, and also 22 patents of exUSSR, Russia and Mexico, and several scientific books.

1. Introduction

We consider motion detection in terms of robust change detection in pixels in an image. The proposed method will not be able to distinguish completely changes due to motion from other changes due to rapidly camera zoom in the video sequence analyzed and movement in the scene present. Despite these problems, there exist numerous applications for this kind of motion detection. It can be used for surveillance objectives, e.g. to monitor a room, in which there is not supposed to be any motion, or the detection results can be useful as input data for more advanced, higher level video processing techniques. There are exist various techniques to detect pixel-by-pixel changes, one of these is to simply substract the color levels of successive frames, and to conclude that the pixel has changed when the outcome exceeds a present threshold. We have developed the mathematical operations to consume less time that can be achieved dividing different operations depending of parameters obtained using fuzzy logic membership functions. This permits to realize robust noise and movement detection.

The main idea is to use an adaptive threshold that is depended on the local pixel statistics and the spatial pixel context. The resulting method is insensitive to a noise; and it is locally adaptive to spatially varying noise levels. This method uses data incoming during long period of time, and the threshold is adapted to both temporal and spatial information [1-4].

The noise should not be labeled as a motion, while on the other hand, it is not so important if not every single changed pixel of an object is detected. However, in the case of motion detection during the denoising processing, where the detection result is used for temporal filtering, undetected changes in an object can lead to motion blur, but in the same time if some of noise is labeled as motion it is no so critical.

Using fuzzy logic techniques we aim at defining a confidence measure with respect to the existence of motion, to be called hereafter "motion confidence" [2].

In the paper, we present the performances of the proposed fuzzy logic algorithm in terms of *PSNR* and *MAE* [4], which characterizes the noise suppression and fine detail and edge preserving abilities. Section 2 exposes the algorithm framework for simultaneous motion detection and video denoising, and the principal blocks of the proposed algorithm. In section 3, we explain the proposed fuzzy logic recursive motion detector applying temporal processing. Section 4 presents simulation results in terms of *PSNR* and *MAE* criteria values, and, also some frames processed to distinguish visually performance of investigated algorithms. Finally, in section 6 the conclusions are given.

2. Framework of the algorithm

2.1. Algorithm structure

The structure of the proposed algorithm is exposed in Fig. 1, where, firstly a noisy frame is processed with fuzzy vectorial motion detection employing a current and a previously processed frame in a sliding window 5x5 to provide reference values for following processing steps. We use the algorithms for Gaussian noise estimator obtaining the parameter, which helps to suppress a noise in a first step. In first instance we have a mean value denoted as, $\bar{y} = \frac{\sum_{i=1}^{N} y_i}{N}$; $i = 1, ..., N; N = 9$; also, an angle deviation for multichannel images (in case of color image RGB space) is calculated in central pixel and mean value to identify deviation in central reference:

$$\theta_c = A\left(\bar{y}, y_c\right), \qquad (1)$$

where y_c = central pixel.

So, in this step the values in the angle range [0, 1.57] are obtained, to avoid a bad estimation of noise presence as follows:

GAUSSIAN ESTIMATION ALGORITHM

Step 1) IF $\theta_c \le F/255$ THEN Histogram is increased in "1", else is "0".

Step 2) Calculate probabilities for each one of these samples:

$$S = \sum_i Histogram_i;\; p_i = \frac{Histogram_i}{S};\; i = 0, ..., 255.$$

Step 3) Obtain standard deviation σ_T':

$$\mu = \sum_{i=0}^{255} i \cdot p_i;\; \sigma^2 = \sum_{i=0}^{255} \left(i - \mu\right)^2 \cdot \left(p_i\right)\; \sigma'_T = \sqrt{\sigma_T^2}\,.$$

Additionally, applying the video sequences in a RGB space format, each plane of the image is processed in an independent manner, and parameters $\sigma_T{}' = \sigma_{red}{}' = \sigma_{green}{}' = \sigma_{blue}{}'$ are adapted along the video sequence. At the first step the angle deviations of central pixel with respect to others in a sliding 3x3 window are found [4]: $\theta_i = A\left(x_i, x_c\right)\; x_i; i = 1, \ldots, N-1;$ where $i \ne central\ pixel$. After that the uniform regions can be detected and filtered applying a mean weighted filtering algorithm [5]:

MEAN WEIGHTED FILTERING ALGORITHM

Step 1) IF $\theta_2\ AND\ \theta_4\ AND\ \theta_7\ AND\ \theta_5 \ge \tau_1$ THEN,

$$y_{out} = \frac{\sum_{\substack{i=1 \\ i \neq c}}^{N-1} x_i \cdot \left(\frac{2}{\left(1+e^{\theta_i}\right)^r} \right) + x_c}{\sum_{i=1}^{N-1} \left(\frac{2}{\left(1+e^{\theta_i}\right)^r} \right) + 1} . \quad (2)$$

Step 2) IF θ_6 *AND* θ_3 *AND* θ_1 *AND* $\theta_8 \geq \tau_1$ THEN,

$$y_{out} = \frac{\sum_{\substack{i=1 \\ i \neq c}}^{N-1} x_i \cdot \left(\frac{2}{\left(1+e^{\theta_i}\right)^r} \right) + x_c}{\sum_{i=1}^{N-1} \left(\frac{2}{\left(1+e^{\theta_i}\right)^r} \right) + 1} . \quad (3)$$

Such the algorithm realizes the fast smoothing for Gaussian noise. Besides, the central pixel has the highest weight with "1" value to preserve some characteristics in the uniform region.

If any of two steps presented before are satisfied, other considerations will be taken to process the noisy sample. We take into account a sliding 5x5 window to process and estimate the standard deviation in same way, as in "*GAUSSIAN ESTIMATION ALGORITHM*". The difference in here is that the frame in a RGB space is separated in each one of the planes to process them in an independent way obtaining values $\sigma_{red}, \sigma_{green}, \sigma_{blue}$: After that we compare each one with $\sigma_{red}', \sigma_{green}', \sigma_{blue}'$, respectively, in order to have a similarity value for each sample and to have a criterion in performance more or less filtering charge. From simulation experiments we obtained that, if $\sigma_T' < \sigma_T$, then $\sigma_T = \sigma_T'$, otherwise $\sigma_T' = \sigma_T$ (parameter $T = red, green, blue$ for each a plane) permitting to improve the quality of temporal filtering stage.

It has been defined experimentally by optimum of *PSNR* and *MAE* criteria the value of a threshold to preserve some important characteristics in a spatially filtered frame that will be used in temporal algorithm: $T_{red} = 2\sigma_T$, where T_{red} is threshold in red plane.

2.2. Fuzzy Vectorial Gradient Values

For each pixel *(i,j)* of the red component image, we use a 3x3 neighborhood window. Each neighbour of *(i,j)* corresponds to one direction: *N = North, E = East, S = South, W = West, NW = North West, NE = North East, SE = South East, SW = South West*.

If A_R denotes the red component input image, then the gradient can be defined as $\nabla_{(k,l)} A_R(i,j) = \left| A_R(i+k, j+l) - A_R(i,j) \right|, k,l \in \{-1,0,1\}$,(4)

where the pair (k,l) corresponds to one of the eight directions that are called *the basic gradient values* [2], and point (i,j) is called *the centre of the gradient*. To avoid blur in the edge presence it is used not only one basic gradient for each direction but also two related gradient values. These three gradient values for a certain direction are finally connected together into one single value called *fuzzy gradient value*. Now we take pixels as the vectors to realize directional processing, taking the same procedure as in gradient values. By this way we obtain *fuzzy vectorial gradient* values that are defined by the *Fuzzy Rule 1*. The two related gradient values in the "same direction" as the basic gradient, are determined by the "centers" making a right-angle with the direction of the corresponding basic gradient [3], as it is illustrated in Fig. 3. Table 1 exposes the involved gradient values.

We can define a variable $\gamma = NW, N, NE, E, SE, S, SW, W$ [7]. By this way it can be used the threshold value obtained before and the gradient values to have fuzzy vectorial gradient values in such a form:

ALGORITHM TO OBTAIN FUZZY VECTORIAL

GRADIENT VALUES IN BASIC GRADIENTS

Step 1) IF $\nabla_{\gamma\beta} < T_{s\beta}$ THEN

calculate angle deviation in this direction $\alpha_{\gamma\beta}$ and obtain weight value $\alpha_{\gamma\beta}' = \frac{2}{\left(1+e^{\alpha_{\gamma\beta}}\right)^r}$. (4)

.*Step 2)* Find the basic vectorial gradient using membership function.

Step 3) IF $\nabla_{\gamma\beta} > T_{s\beta}$ THEN $\mu_{BIG} = 0$.

where, $\beta = red, gree, blue$ and $r = 1$ channels in frame in video sequence. Membership function is defined as:

$$\mu_{BIG} = \max(x, y) , \quad , \quad (5)$$

where $x = \alpha_{\gamma\beta}'$ and $y = \left(1 - \nabla_{\gamma\beta} / T_{s\beta}\right)$. To obtain a significant angle deviation in each a plane of the image we use the angle formed by vectors according to [5].

$$\alpha = \cos^{-1}\left(\frac{r_1 \cdot r_2 + g_1 \cdot g_2 + b_1 \cdot b_2}{\sqrt{r_1^2 + g_1^2 + b_1^2} \cdot \sqrt{r_2^2 + g_2^2 + b_2^2}} \right), \quad (6)$$

where, (r_1, g_1, b_1) and (r_2, g_2, b_2) are coordinates of two color pixels.

Last algorithm describes the process necessary to determine basic vectorial gradient value. To determine related vectorial gradients the procedure is similar:

ALGORITHM FOR FUZZY VECTORIAL GRADIENT VALUES IN RELATED GRADIENTS

Step 1) IF $\nabla_{\gamma\beta(R1,R2)} < T_{s\beta}$ THEN calculate angle deviation in this direction $\alpha_{\gamma\beta(R1,R2)}$ and obtain weight value [6]

$$\alpha_{\gamma(R1,R2)\beta}' = \frac{2}{\left(1+e^{\alpha_{\gamma\beta(R1,R2)}}\right)^r}, \qquad (7)$$

Step 2) Find the related vectorial gradient using membership function.

Step 3) IF $\nabla_{\gamma\beta(R1,R2)} > T_{s\beta}$ THEN $\mu_{BIG} = 0$. $(R1, R2)$ are the related vectorial gradients found in each a direction as it is shown in Fig. 3. Membership function $\mu_{BIG} = \max(x, y)$ is defined as $x = \alpha_{\gamma\beta(R1,R2)}'$ and $y = \left(1 - \nabla_{\gamma\beta(R1,R2)} / T_{s\beta}\right)$. (8)

"Fuzzy vectorial gradient values" can be found using *Fuzzy Rule 1,* where basic and related vectorial gradient values for each direction are connected together.

2.3. Fuzzy Rule 1.

Defining the fuzzy vectorial gradient value $\nabla^F_{\gamma\beta} A_\beta(i,j)$

IF $\nabla_{\gamma\beta}$ is BIG AND $\nabla_{\gamma\beta R1}$ is BIG, OR $\nabla_{\gamma\beta}$ is BIG AND $\nabla_{\gamma\beta R2}$ is BIG, THEN $\nabla^F_{\gamma\beta} A_\beta(i,j)$ is BIG,

where $\nabla_{\gamma\beta}$ represents the basic vectorial gradient value and $\nabla_{\gamma\beta R1}$ and $\nabla_{\gamma\beta R2}$ represents the two related vectorial gradient values for the direction γ in the channel β.

Observing the *Fuzzy Rule 1,* it is easy to see following: suppose that basic and related vectorial gradients are close enough in any way, in absolute difference (absolute norm) or in a vectorial criterion in angle distances (that's why we change gradient values to vectorial gradient values), so, this proposal is developed to obtain parameters robustness, giving a better understanding of the nature of pixels in a window processing. Under this criterion we will have values denoted as fuzzy vectorial gradients that means nearby in pixels related, and they are helpful to suppress Gaussian noise corruption presented on the sample,. So, suppression is done by a weighted mean procedure where nearby close to 1 have the bigger weights in the algorithm due to the nature of the proposed procedure used in membership function. This suppresses noise more efficiently but smooth details and edges, in our complete algorithm, the temporal filtering are designed. The reference values where found modifying their parameters according to optimum of PSNR and MAE criteria values. Spatial algorithm presents good results in suppression noise compared with some other algorithms found in literature [1-3].

The weighted mean algorithm is implemented by the next expression:

$$y_{out} = \frac{\sum_{i=0}^{N-1} y_\gamma \cdot x_{\gamma i}}{\sum_{i=0}^{N-1} x_{\gamma i}}; \quad \gamma = NW, N, NE, E, SE, S, SW, W \qquad (9)$$

Mean value is found doing multiplication of fuzzy vectorial gradient value with his respective pixel in that direction γ, omitting central pixel.

All this procedures are repeated in same manner to green and blue channels.

3. Temporal filtering algorithm

The reference values of spatial filter presented above are used in the final step in proposed filter.

In here only the past and present frames are used to avoid dramatic charge in memory requirements and time processing. The fuzzy logic rules are used in each a plane of two frames in independent manner. Firstly, we found different parameters in each pixel position of a 3x3 window processing to adequate our algorithm with parameters related.

Let us find the angle deviations and gradient values for a central pixel in present frame respect to its neighbours in past frame, all should be done for each plane of the frames.

$$\theta_i^1 = A\left(x_i^A, x_c^B\right), i = 1, \ldots, N; N = 9,$$

$$\nabla_i^1 = \left|x_i^A - x_c^B\right|; i = 1, \ldots, N;\ N = 9, \qquad (10)$$

where x_c^B is central pixel in present frame, and *A* and *B* letters represent past and present frames by planes respectively.

Applying (10) we can find the angle and gradient values for the corresponding pixel positions in each pixel of two frames, and also only for the present frame, eliminating operations in past frame.

Let us define the membership functions used to obtain a value that indicates the degree, in which a certain gradient value or vectorial value matches the predicate. If a gradient or a vectorial value have membership degree one, for the fuzzy set SMALL, it means that it is SMALL for sure in this fuzzy set. Selection of this kind of membership functions is follow from nature of pixels, where a movement is not a linear response, and a pixel has different meanings in each scene of the video sequence.

Membership functions BIG and SMALL for angles and gradients are given by next expressions [8]:

$$\mu_{SMALL}(\theta)=\begin{cases}1 & if\ \theta<med1\\ \exp\left\{-\left(\dfrac{(\theta-med1)^2}{2\cdot\sigma^2}\right)\right\} & otherwise\end{cases}, \quad (11a)$$

$$\mu_{SMALL}(\nabla)=\begin{cases}1 & if\ \nabla<med2\\ \exp\left\{-\left(\dfrac{(\nabla-med2)^2}{2\cdot\sigma^2}\right)\right\} & otherwise\end{cases}, \quad (11b)$$

$$\mu_{BIG}(\theta)=\begin{cases}1 & if\ \theta>med1\_1\\ \exp\left\{-\left(\dfrac{(\theta-med1\_1)^2}{2\cdot\sigma^2}\right)\right\} & otherwise\end{cases}, \quad (11c)$$

$$\mu_{BIG}(\nabla)=\begin{cases}1 & if\ \nabla>med2\_2\\ \exp\left\{-\left(\dfrac{(\nabla-med2\_2)^2}{2\cdot\sigma^2}\right)\right\} & otherwise\end{cases}, \quad (11d)$$

where $med1=0.2$, $med2=60$, $med1\_1=0.9$, and $med2\_2=140$, using $\sigma^2=0.1$ by med1 and med2, and using $\sigma^2=1000$ by med1_1 and med2_2.

Now we use Fuzzy Rules 2, 3, 4, and 5 to acquire corresponding values:

Fuzzy Rule 2: Defining the fuzzy gradient-vectorial value $SBB(x,y,t)$.

IF $\theta^1(x,y,t)$ is SMALL AND $\theta^2(x,y,t)$ is BIG AND $\theta^3(x,y,t)$ is BIG AND $\nabla^1(x,y,t)$ is SMALL AND $\nabla^2(x,y,t)$ is BIG AND $\nabla^3(x,y,t)$ is BIG THEN $SBB(x,y,t)$ is true.

Fuzzy Rule 3: Defining the fuzzy gradient-vectorial value $SSS(x,y,t)$.

IF $\theta^1(x,y,t)$ is SMALL AND $\theta^2(x,y,t)$ is SMALL AND $\theta^3(x,y,t)$ is SMALL AND $\nabla^1(x,y,t)$ is SMALL AND $\nabla^2(x,y,t)$ is SMALL AND $\nabla^3(x,y,t)$ is SMALL THEN $SSS(x,y,t)$ is true.

Fuzzy Rule 4: Defining the fuzzy gradient-vectorial value $BBB(x,y,t)$.

IF $\theta^1(x,y,t)$ is BIG AND $\theta^2(x,y,t)$ is BIG AND $\theta^3(x,y,t)$ is BIG AND $\nabla^1(x,y,t)$ is BIG AND $\nabla^2(x,y,t)$ is BIG AND $\nabla^3(x,y,t)$ is BIG THEN $BBB(x,y,t)$ is true.

Fuzzy Rule 5: Defining the fuzzy gradient-vectorial value $BBS(x,y,t)$.

IF $\theta^1(x,y,t)$ is BIG AND $\theta^2(x,y,t)$ is BIG AND $\theta^3(x,y,t)$ is SMALL AND $\nabla^1(x,y,t)$ is BIG AND $\nabla^2(x,y,t)$ is BIG AND $\nabla^3(x,y,t)$ is SMALL THEN $BBS(x,y,t)$ is true.

Where $\theta^1(x,y,t)$, $\theta^2(x,y,t)$, and $\theta^3(x,y,t)$ are the angles and $\nabla^1(x,y,t)$, $\nabla^2(x,y,t)$, and $\nabla^3(x,y,t)$ are the gradient values obtained.

If $SBB(x,y,t)$ is the biggest value found among the others, we realize the following procedure:

ALGORITHM TO FUZZY RULE $SBB(x,y,t)$

Step 1) IF {($SBB(x,y,t)>SSS(x,y,t)$) AND ($SBB(x,y,t)>BBB(x,y,t)$) AND ($SBB(x,y,t)>BBS(x,y,t)$)} THEN Weighted mean is defined as

$$SBB(x,y,t),\ y_{out}=\frac{\sum p^A(x,y,t)\cdot SBB(x,y,t)}{\sum SBB(x,y,t)}, \quad (12)$$

where, $p^A(x,y,t)$, represents each a pixel in the last frame that fulfills with the IF condition, and y_{out} is the outputs filtered in spatial and temporal filtering.

Step 2) Update standard deviation for next frames to divide fine details from uniform regions

To update standard deviation we need an expression that varies in agree with parameters developed in each a sample of the image, that is why we should have different values by each a condition in our algorithm to characterize in an independent manner each region of the image. This can be achieved using the below expression after by each Fuzzy Rule to update the parameter:

$$\sigma_T{}'=(\alpha\cdot\sigma_{TOTAL})+(1-\alpha)\cdot(\sigma_T{}'), \quad (13)$$

where, $T=red, green, blue$, α for the $SBB(x,y,t)$ fuzzy value will be equal to 0.875, and σ_{TOTAL} is defined by $\sigma_{TOTAL}=(\sigma_{red}+\sigma_{green}+\sigma_{blue})/3$.

The $SBB(x,y,t)$ value says that a central pixel is in movement because of big differences in corresponding local and gradient values.

If $SSS(x,y,t)$ is the biggest value among the other ones, we should use the following procedure.

ALGORITHM TO FUZZY RULE $SSS(x,y,t)$

ISBN 978-3-901608-32-2

Step 1) IF {($SSS(x,y,t) > SBB(x,y,t)$) AND ($SSS(x,y,t) > BBB(x,y,t)$) AND ($SSS(x,y,t) > BBS(x,y,t)$)} THEN Weighted mean using $SSS(x,y,t)$ as the weights is defined:

$$y_{out} = \frac{\sum (p^A(x,y,t)\cdot 0.5 + p^B(x,y,t)\cdot 0.5)\cdot SSS(x,y,t)}{\sum SSS(x,y,t)}, \quad (14)$$

where, $p^A(x,y,t)$ and $p^B(x,y,t)$ represent each a pixel in last and present frames that satisfies to the IF condition, and y_{out} is the filtered outputs.

Step 2) Update standard deviation for next frames to divide fine details from uniform regions.

Only pixels that obey the IF conditions will be taken into account to calculate the weighted mean. The value α for the $SSS(x,y,t)$ fuzzy value will be equal to 0.1255. The $SSS(x,y,t)$ value shows that a central pixel is not in movement because of small differences in all directions, that is why we use the pixels in both frames.

If $BBB(x,y,t)$ is the biggest value among the other ones, we can conclude that majority of pixels are not related in any way with the other neighborhood pixels. Here it is decided to realize the following procedure.

ALGORITHM TO FUZZY RULE $BBB(x,y,t)$

Step 1) IF {($BBB(x,y,t) > SBB(x,y,t)$) AND ($BBB(x,y,t) > SSS(x,y,t)$) AND ($BBB(x,y,t) > BBS(x,y,t)$)} THEN motion-noise = true.

To solve this problem, consider the nine fuzzy gradient-vectorial values obtained from $BBB(x,y,t)$, and take the central value and at least three fuzzy neighbors values more to detect movement present in the sample. We use the Fuzzy Rule "R" to obtain motion-noise confidence. The activation degree of "R" is just the conjunction of the four subfacts, which are combined by a chosen triangular norm defined as $A\ AND\ B = A * B$. Computations are specifically the intersection of all possible combinations of $BBB(x,y,t)$ and three different neighboring BIG membership degrees $BBB(x+i, y+j, t)$, $(i, j = -1,0,1)$, using triangular norm. This can give 56 different values, which should be summed using algebraic sum of all instances to obtain the motion-noise confidence. Algebraic sum is given by $A\ OR\ B = A + B - A * B$.

Step 2) If $\sqrt{motion-noise\ confidence} = 1$

then $\alpha = 0.875$,

else if $\sqrt{motion-noise\ confidence} = 0$

then $\alpha = 0.125$

else $\alpha = 0.5$.

Step 3)

$$y_{out} = (1-\alpha)\cdot(pres\_fr_{central\_pixel}) + \alpha\cdot(past\_fr_{central\_pixel}). \quad (15)$$

The $BBB(x,y,t)$ value shows that a central pixel and its neighbors do not have relation among the others and it is highly probably that this pixel is in motion or is a noisy pixel.

If $BBS(x,y,t)$ is the biggest value found from the other ones, we realize the follow procedure:

ALGORITHM TO FUZZY RULE $BBS(x,y,t)$

Step 1) IF {($BBS(x,y,t) > SBB(x,y,t)$) AND ($BBS(x,y,t) > SSS(x,y,t)$) AND ($BBS(x,y,t) > BBB(x,y,t)$)} THEN Weighted mean using $BBS(x,y,t)$ as the weights is defined:,

$$y_{out} = \frac{\sum p^B(x,y,t)\cdot(1-BBS(x,y,t))}{\sum(1-BBS(x,y,t))}, \quad (16)$$

where, $p^B(x,y,t)$ represents each a pixel in present frame that satisfies to the IF condition, and y_{out} is the filtered output after spatial and temporal processing.

Step 2) Update standard deviation for next frames to divide fine details from uniform regions.

Now it can be applied the *Spatial Filter* to smooth the non-stationary noise left by the preceding temporal filter. This is done by a local spatial filter, which adapts to image structures and noise levels present in the corresponding spatial neighborhood. The spatial filter is the same that has been explained in sec. 2.

4. Experimental results

Here, we present the performance of the proposed algorithm obtained by simulation of the novel and existed techniques (see Figs.3, 4 and Tab. 2). It has been investigated two different video sequences to qualify effectiveness of this filter. Both sequences: "Miss America" and "Flowers" present different image texture to provide a better understanding in the robustness of the fuzzy logic algorithm. Video sequences were contaminated with different Gaussian noise levels, from 0.00 to 0.05 in variance with zero mean. Frames are treated in an RGB color space with 24 bits (true color), 8 bits for each channel, 176x144 pixels in a QCIF format, and size of 100 frames for each a video sequence. The filtered frames were evaluated in the terms of *PSNR* and *MAE* criteria.

The proposed *Fuzzy Directional Adaptive Recursive Temporal Filter* (*FDARTF_G*) was compared with another similar algorithm, the *FMRSTF* [1-3] (*Fuzzy Motion Recursive Spatial-Temporal Filter*), which works only with gradients, and with an adaptation to this algorithm that employs the angle deviations *FVMRSTF (Fuzzy Vectorial Motion Recursive Spatio-Temporal Filte*), which was not published yet. The Fig. 3 demonstrates the visual performance of the algorithms, and the Fig.4 and Table 2 exposes the objective criteria PSNR and MAE confirming better properties of the proposed *FDARTF_G* algorithm.

5. Conclusions

In this paper a novel robust adaptive recursive scheme for fuzzy logic based motion detection that applies gradients and vectors is presented. The proposed algorithm works in a closed loop realizing the spatial and temporal filtering to improve suppression noise performance and preservation of fine details and edges. It is demonstrated that taking into account, both characteristics (gradients and vectors) and connecting them together, we can realize an algorithm with better performance, improving the techniques that use such characteristics in a separate form. In future, this idea will be extended to suppress impulsive random noise in multichannel filtering.

Acknowledgment. This work was supported by National Polytechnic Institute of Mexico and CONACYT.

6. References

[1] [1] Vladimir Zlokolica, Matthias De Geyter, Stefan Schulte, Aleksandra Pizurica, Wilfried Philips, and Etienne Kerre, Fuzzy Logic Recursive Motion Detection for Tracking and Denoising of Video Sequences. IS&T/SPIE Symposium on Electronic Imaging, San Jose, California, USA, January 2005.

[2] [2] Vladimir Zlokolica. Advanced Non-Linear Methods for Video Denoising. PhD Thesis, Gent University, 2006.

[3] [3] V. Zlokolica, S. Schulte, A. Pizurica, W. Philips, E. Kerre, Fuzzy Logic Recursive Motion Detection and denoising of Video Sequences. *Journal of Electronic Imaging,* 15(2), 2006.

[4] [4] V. Ponomaryov, A. Rosales-Silva; V. Golikov Adaptive and vector directional processing applied to video colour images. *Electronics Letters*, 42(11): 623-624, 2006.

[5] [5] P. E. Trahanias and A. N. Venetsanopoulos, “Vector Directional Filters-A New Class of Multichannel Image Processing Filters”, *IEEE Transactions on Image Processing,* 2(4), 1993.

[6] [6] K. N. Plataniotis and A. N. Venetsanopoulos, Color Image Processing and Applications, Springer Verlag, Berlin, 2000.

[7] [7] Stefan Schulte, Valérie De Witte, Mike Nachtegael, Dietrich Van der Weken, and Etienne E. Kerre. Fuzzy Two-Step Filter for Impulse Noise Reduction From Color Images. *IEEE Transactions on Image Processing,* 15(11), 2006.

[8] [8] Fuzzy Logic Fundamentals, Chapter 3, pp. 61-103, 2001, www.informit.com/content/images/0135705991/samplechapter/0135705991.pdf.

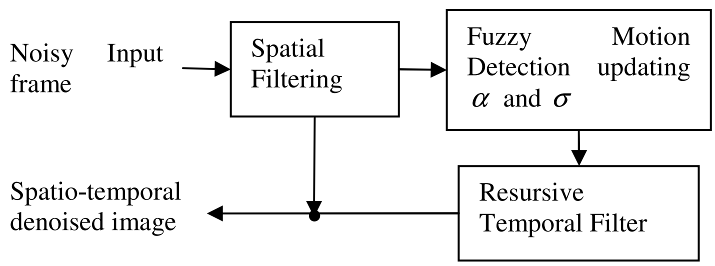

Fig. 1 Proposed denoising scheme using spatial-temporal techniques

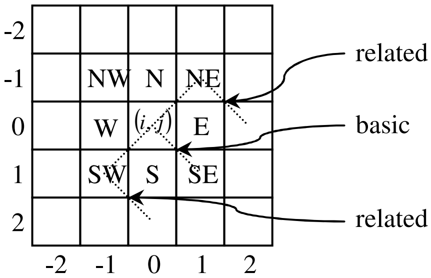

Fig. 2 Basic, related and vectors gradients

Tab.1 Involved gradient values to calculate the fuzzy vectorial gradient

Direction	Basic Gradient Involved	Related Gradients Involved
NW	$(i, j), (i-1, j-1)$	$(i+1, j-1), (i-1, j+1)$
N	$(i, j), (i-1, j)$	$(i, j-1), (i, j+1)$
NE	$(i, j), (i-1, j+1)$	$(i-1, j-1), (i+1, j+1)$
E	$(i, j), (i, j-1)$	$(i-1, j), (i+1, j)$
SE	$(i, j), (i, j+1)$	$(i-1, j+1), (i+1, j-1)$
S	$(i, j), (i+1, j-1)$	$(i, j-1), (i, j+1)$
SW	$(i, j), (i+1, j)$	$(i-1, j-1), (i+1, j+1)$
W	$(i, j), (i+1, j+1)$	$(i-1, j), (i+1, j)$

a) b) c)

Fig. 3 100th Flowers frame; a) Column images by *FLRSTF_NORMAL*, b) Column images by *FLRSTF_ANGLE*, and c) Column images by *FDARTF_G*. These images were filtered from Gaussian noise corrupted ones with variance 0.001, 0.01, 0.015, 0.02, and 0.03 from top to bottom

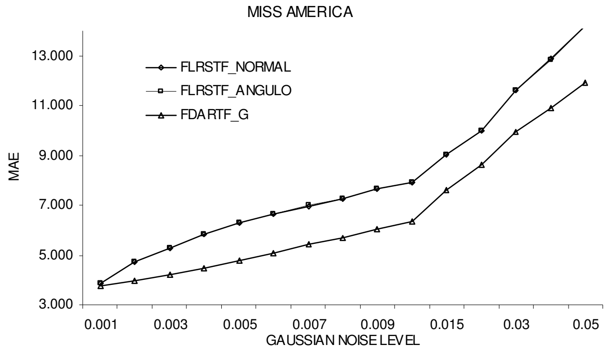

Fig. 4 MAE Values for Miss America Frame No.100, with different Gaussian noise levels of corruption

Tab. 2 PSNR values

PSNR IN MISS AMERICA FRAME 100			
Gaussian Noise Level	FLRSTF_ NORMAL	FLRSTF_ ANGULO	FDARTF_G
0.001	**33.765**	33.758	33.459
0.002	31.912	31.906	**33.106**
0.003	30.860	30.849	**32.777**
0.004	29.963	29.976	**32.371**
0.005	29.283	29.292	**31.829**
0.006	28.791	28.798	**31.383**
0.007	28.332	28.337	**30.875**
0.008	27.977	27.978	**30.504**
0.009	27.538	27.550	**30.085**
0.01	27.259	27.252	**29.613**
0.015	26.062	26.078	**28.104**
0.02	25.163	25.153	**26.950**
0.03	23.859	23.871	**25.701**
0.04	22.872	22.876	**24.941**
0.05	21.986	21.986	**24.231**

