Fuzzy Vector Directional Filters for Multichannel Image Denoising

Alberto Rosales-Silva[1], Volodymyr I. Ponomaryov[1], and Francisco J. Gallegos-Funes[2]

National Polytechnic Institute of Mexico,
Mechanical and Electrical Engineering Higher School
[1] ESIME-Culhuacan; Av. Santa Ana 1000, Col. San Francisco Culhuacan,
04430, Mexico D.F., Mexico
vponomar@ipn.mx
[2] ESIME-Zacatenco; Av. IPN s/n, U.P.A.L.M. Col. Lindavista,
07738, Mexico D.F., Mexico
fgallegosf@ipn.mx

Abstract. We propose a fuzzy logic recursive scheme using directional processing for motion detection and spatial-temporal filtering to decrease Gaussian noise corruption. We introduce novel ideas that employ the differences between images. That permits to connect these using angle deviations in them obtaining several parameters and applying them in the robust algorithm that is capable to detect and differentiate movement in background of noise in any way.

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

1 Introduction

We consider motion detection in terms of robust change detection in pixels in an image [1]. 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 [1-3]. Some techniques 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 suppression and movement detection.

The main idea is to use adaptive threshold that is adapted to the local pixel statistics and the spatial pixel context. The proposed method is insensitive to noise; it is locally adaptive to spatially varying noise levels. The presented method uses data incoming during long period of time, and the threshold is adapted to both temporal and spatial information [1-3].

The noise should not be labelled as motion, it is not so important if not every single changed pixel of an object is detected. However, in the case of motion detection

L. Rueda, D. Mery, and J. Kittler (Eds.): CIARP 2007, LNCS 4756, pp. 124–133, 2007.

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 noise is labelled 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].

2 Framework Method

We expose the framework of the algorithm for simultaneous motion detection and video denoising. The proposed algorithm is shown in Figure 1 where a noisy frame is processed firstly with fuzzy vector motion detection using the current and previous processed frames using a 5x5 window to provide the reference values for following processing stages.

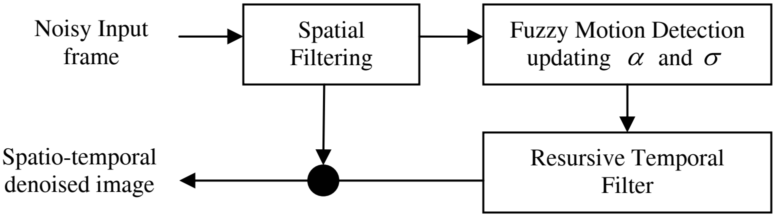
Noisy Input frame
Spatial Filtering
Fuzzy Motion Detection updating α and σ
Spatio-temporal denoised image
Resursive Temporal Filter

Fig. 1. Proposed denoising scheme using spatial-temporal techniques

2.1 Spatial Filtering

We use the Gaussian estimation algorithm to suppress the noise in the first stage,
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:

$$S=\sum_i Histogram_i\ ;\ p_i = Histogram_i/S\ ,\ i=0,\ldots,255. \tag{1}$$

Step 3) Obtain standard deviation σ'_T :

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

where $\theta_c = A\left(\bar{y}, y_c\right)$ is the angle deviation, $\bar{y} = {}^1\!/_N \sum_{i=1}^{N} y_i$ is the mean value with i=1,…,N, N=9, and y_c is the central pixel.

Additionally, each plane of the video sequence in RGB space format is processed in an independent way, and the parameters $\sigma'_T = \sigma'_{red} = \sigma'_{green} = \sigma'_{blue}$ are adapted along the video sequence. The angle deviations of central pixel with respect to others in a 3x3 window are found [4]: $\theta_i = A\left(x_i, x_c\right)$, $x_i; i=1,\ldots,N-1$; where $i \ne central\ pixel$.

Now, we detect uniform regions using the mean weighted filtering algorithm [5]:
Step 1) IF θ_2 *AND* θ_4 *AND* θ_7 *AND* $\theta_5 \geq \tau_1$ THEN,

$$y_{out} = \sum_{\substack{i=1 \\ i \neq c}}^{N-1} x_i \left[\frac{2}{\left(1+e^{\theta_i}\right)^r} \right] + x_c \Bigg/ \sum_{i=1}^{N-1} x_i \left[\frac{2}{\left(1+e^{\theta_i}\right)^r} \right] + 1 \qquad (3)$$

Step 2) IF θ_6 *AND* θ_3 *AND* θ_1 *AND* $\theta_8 \geq \tau_1$ THEN, Use Eq. (3).

Such the algorithm can realize the smoothing fast for Gaussian noise. Besides, the central pixel has the highest weight with "1" value to preserve some characteristics in uniform regions. We use a 5x5 processing window to estimate the standard deviation in same way as in the Gaussian estimation algorithm to obtain the values σ_T, and then we compare with the values σ'_T, in order to have a similarity value for each sample and to have a criterion in perform more or less filtering charge. From experiments we obtained that if $\sigma'_T < \sigma_T$, then $\sigma_T = \sigma'_T$, otherwise $\sigma'_T = \sigma_T$, where T can be the component red, green, or blue, that permits to improve temporal filtering algorithm. It has been defined experimentally by optimum PSNR and MAE criteria a threshold $Th_T = 2\sigma_T$ to preserve some important characteristics in a spatially filtered frame that will be used in temporal algorithm.

2.2 Fuzzy Vector Gradient Values

For each pixel *(i,j)* of the any component image, we use a 3x3 neighbourhood window. Each neighbour of *(i,j)* corresponds to one direction as illustrates Table 1.

If A_T denotes one component input image, the definition of the gradient can be defined as $\nabla_{(k,l)} A_T(i,j) = \left| A_T(i+k, j+l) - A_T(i,j) \right|$ with $k,l \in \{-1,0,1\}$, where the pair *(k,l)* corresponds to one of the eight directions that are called *the basic gradient values* [2], and *(i,j)* is called the centre of the gradient. To avoid blurred in presence of an edge, it is used not only one basic gradient for each direction but also two related gradient values. The 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 have directional process, taking the same procedure as in gradient values. By this way we obtain Fuzzy vector gradient values that are defined by the *Fuzzy Rule 1*. The two related gradient values in the same direction and the basic gradients are determined by the centres making a right-angle with the direction of the corresponding basic gradient [3].

Table 1. Involved gradient values to calculate the fuzzy vector 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*)

We can use the threshold Th_T obtained before and use the gradient values to have fuzzy vector gradient values in such a form,

Step 1) IF $\nabla_{\gamma\beta} < T_{s\beta}$ THEN calculate angle deviation in the direction $\alpha_{\gamma\beta}$ and obtain weight value,

$$\alpha'_{\gamma\beta} = 2 \Big/ \left(1 + e^{\alpha_{\gamma\beta}}\right)^r \tag{4}$$

Step 2) Obtain basic vector gradient using membership function.
Step 3) IF $\nabla_{\gamma\beta} > T_{s\beta}$ THEN $\mu_{BIG} = 0$.

where $T_{s\beta} = Th_T$, $\gamma = NW, N, NE, E, SE, S, SW, W$ [6], $\beta = red, gree, blue$ and $r = 1$ channels in video sequence, the membership function is $\mu_{BIG} = \max(x, y)$ with $x = \alpha'_{\gamma\beta}$ and $y = \left(1 - \nabla_{\gamma\beta} / T_{s\beta}\right)$. To obtain the angle deviation in each plane of the image we select to work in the angle formed by vectors in only one coordinate [4]:

$$\alpha = \cos^{-1}\left(\left(r_1 r_2 + g_1 g_2 + b_1 b_2\right) \Big/ \sqrt{\left(r_1^{\ 2} + g_1^{\ 2} + b_1^{\ 2}\right)\left(r_2^{\ 2} + g_2^{\ 2} + b_2^{\ 2}\right)}\right) \tag{5}$$

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

Last algorithm describes the process necessary to determine the basic vector gradient value. To determine related vector gradients the procedure is the following:
Step 1) IF $\nabla_{\gamma\beta(R1,R2)} < T_{s\beta}$ THEN compute angle deviation in direction $\alpha_{\gamma\beta(R1,R2)}$ and obtain weight value [5],

$$\alpha'_{\gamma\beta(R1,R2)} = 2 \Big/ \left(1 + e^{\alpha_{\gamma\beta(R1,R2)}}\right)^r \tag{6}$$

Step 2) Obtain related vector gradient using membership function.
Step 3) IF $\nabla_{\gamma\beta(R1,R2)} > T_{s\beta}$ THEN $\mu_{BIG} = 0$.

where $(R1, R2)$ are the related vector gradients and the membership function is $\mu_{BIG} = \max(x, y)$ with $x = \alpha'_{\gamma\beta(R1,R2)}$ and $y = \left(1 - \nabla_{\gamma\beta(R1,R2)} / T_{s\beta}\right)$.

The fuzzy rule 1 is defined as,

Fuzzy Rule 1: defining the fuzzy vector 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 $\nabla_{\gamma\beta}$ is the basic vector gradient value, and $\nabla_{\gamma\beta R1}$ and $\nabla_{\gamma\beta R2}$ are two related vector gradient values for the direction γ in the channel β.

If basic and related vector gradients are close enough, in absolute difference (absolute norm) or in a vector criterion in angle distances (that is why we change gradient values to vector gradient values), this proposal is developed to obtain robust parameters, giving a better understanding of the nature of pixels in a window processing. Under this criterion we will have values denoted as fuzzy vector 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 proposed procedure used in membership function. This suppresses noise more efficiently but smoothes details and edges, in our complete algorithm the temporal filtering are designed. The reference values where found modifying their parameters according to optimum PSNR and MAE values. Spatial algorithm presents good results in noise suppression compared with some algorithms found in literature [1-3].

The weighted mean algorithm is implemented by:

$$y_{out} = \sum_{\substack{i=0 \\ i \neq c}}^{N-1} y_{\gamma} \cdot x_{\gamma i} \Bigg/ \sum_{i=0}^{N-1} x_{\gamma i} \tag{7}$$

where mean value is found doing multiplication of fuzzy vector gradient value with his respective pixel in that direction γ.

2.3 Temporal Filtering Algorithm

In here, we explain the proposed fuzzy logic recursive motion detector with temporal algorithm. The reference values of spatial filter presented above are used in the final stage in the proposed filter. 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 plane of two frames in independent way.

We found angle deviations and gradient values by the central pixel in present frame respect to his neighbours in past frame, all done by each plane of the frames by using 3x3 window,

$$\theta_i^1 = A\left(x_i^A, x_c^B\right) \;;\; \nabla_i^1 = \left|x_i^A - x_c^B\right|; i = 1,\dots,N;\; N = 9 \tag{8}$$

where x_c is central pixel in present frame, and A and B are past and present frames by planes, respectively.

Let us define the membership functions used to obtain a value that indicates the degree, in which a certain gradient value or vector value matches the predicate. If a gradient or a vector 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 frame of video sequence.

Membership functions SMALL and BIG for angles and gradients are given by [7]:

$$\mu_{SMALL}(M) = \begin{cases} 1 & M < med \\ \exp\left(-\dfrac{(M - med)^2}{2\sigma_S^2}\right) & \text{otherwise} \end{cases} \tag{9}$$

$$\mu_{BIG}(M) = \begin{cases} 1 & M < med \\ \exp\left(-\dfrac{(M - med)^2}{2\sigma_B^2}\right) & \text{otherwise} \end{cases} \tag{10}$$

where $\sigma_S^2 = 0.1$, $\sigma_B^2 = 1000$, and M can be the angle θ or gradient ∇, for angles *med*=0.2 and *med*=0.9 for membership functions SMALL and BIG, respectively, and for gradients *med*=60 and *med*=140 for SMALL and BIG functions, respectively.

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

Fuzzy Rules
Fuzzy Rule 2: Defining the fuzzy gradient-vector 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-vector 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-vector 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-vector 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 $\theta^r(x,y,t)$ are angles values, $\nabla^r(x,y,t)$ are gradient values, and $r = 1,2,3$.

From the result values of each fuzzy rule (2 – 5) we can compare these values in the following way:

Algorithm to Fuzzy Rule *SBB*(*x*,*y*,*t*)
If *SBB*(*x*,*y*,*t*) is the biggest value found from the others: *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 using *SBB*(*x*,*y*,*t*), $y_{out} = \dfrac{\sum p^A(x,y,t) \cdot SBB(x,y,t)}{\sum SBB(x,y,t)}$ *Step 2)* Update standard deviation for next frames to divide details from uniform regions.

where *SBB*(*x*,*y*,*t*) value says that central pixel is in movement because of big differences in corresponding local and gradient values, $p^A(x,y,t)$ represents each pixel in last frame that fulfil with the IF condition, and y_{out} is the output filtered in spatial and temporal filtering.

To update standard deviation we need different values by each condition in our algorithm to characterize in an independent manner each region of the image. This is achieved using the expression above, and this expression is always used after by each Fuzzy Rule to update the parameter:

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

where $T = red, green, blue$, $\alpha = \alpha_{SBB} = 0.875$, and $\sigma_{TOTAL} = (\sigma_{red} + \sigma_{green} + \sigma_{blue})/3$.

Algorithm to Fuzzy Rule *SSS*(*x,y,t*)

If *SSS*(*x,y,t*) is the biggest value found from the others:

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*),

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

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

where *SSS*(*x,y,t*) shows that a central pixel is not in movement because of small differences in all directions, that is why we use pixels in both frames, $p^A(x,y,t)$ and $p^B(x,y,t)$ are the pixels in last and present frames that fulfil with the IF condition will be taken in count to calculate the weighted mean, $\alpha = \alpha_{SSS} = 0.1255$, and y_{out} is the filtered output.

Algorithm to Fuzzy Rule *BBB*(*x,y,t*)

If *BBB*(*x,y,t*) is the biggest value found from the others:

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.

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

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) $y_{out} = (1-\alpha)\cdot\left(pres\_fr_{central\_pixel}\right) + \alpha\cdot\left(past\_fr_{central\_pixel}\right)$

The *BBB*(*x,y,t*) value shows that a central pixel and its neighbours do not have relation among the others and it is highly probably that this pixel is in motion or is a noisy pixel. To solve this problem, consider the nine fuzzy gradient-vector values obtained from *BBB*(*x,y,t*) and take the central value and at least three fuzzy neighbours 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 neighbouring BIG membership degrees *BBB*(*x*+1,*y*+1,*t*), $(i, j = -1,0,1)$, using triangular norm. This can give 56 different values, which should be summed using algebraic sum $A\ OR\ B = A + B - A*B$ of all instances to obtain the motion-noise confidence.

Algorithm to Fuzzy Rule *BBS*(*x,y,t*)

If *BBS*(*x,y,t*) is the biggest value found from the others:

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*),

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

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

where, $p^{B}\left(x,y,t\right)$ represents each pixel in present frame that fulfil with the IF condition, and y_{out} is the output filtered in spatial and temporal filtering.

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 (see sec. 2.1), which adapts to image structures and noise levels present in the corresponding spatial neighbourhood.

3 Experimental Results

We present the performance of the proposed algorithm obtained by simulation of the proposed techniques. We use "Flowers" and "Miss America" video sequences to qualify effectiveness of this filter 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, 8 bits for each channel, 176x144 pixels in a QCIF format with 100 frames.

The proposed Fuzzy Directional Adaptive Recursive Temporal Filter (FDARTF) was compared with the Vector Median M-type K-nearest Neighbor Filter (VMMKNNF) [8,9] and the Generalized Vector Directional Filter (GVDF) [10,11]. Therefore, the proposed FDARTF filter was compared with other similar algorithms, the Fuzzy Motion Recursive Spatial-Temporal Filter (FMRSTF) [1-3], which works only with gradients, and with an adaptation to this algorithm using angle deviations Fuzzy Vectorial Motion Recursive Spatial-Temporal Filter (FVMRSTF), which was not published yet.

Table 2 presents the performance results in terms of PSNR for the frame #100 of video sequence "Miss America" by use different filters. From the Table 1, one can see that the best results in PSNR criterion are given by the proposed filter.

Table 2. PSNR values for a frame of video sequence "Miss America" by use different filters

Gaussian Noise	VMMKNNF	GVDF	FMRSTF	FVMRSTF	FDARTF
0.001	32.500	33.981	33.765	33.758	33.459
0.002	31.853	32.919	31.912	31.906	33.106
0.003	31.163	32.134	30.860	30.849	32.777
0.004	30.535	31.408	29.963	29.976	32.371
0.005	29.917	30.794	29.283	29.292	31.829
0.006	29.431	30.301	28.791	28.798	31.383
0.007	28.903	29.816	28.332	28.337	30.875
0.008	28.467	29.419	27.977	27.978	30.504
0.009	28.043	29.060	27.538	27.550	30.085
0.01	27.702	28.697	27.259	27.252	29.613
0.015	26.152	27.151	26.062	26.078	28.104
0.02	24.955	25.969	25.163	25.153	26.950
0.03	23.339	24.256	23.859	23.871	25.701
0.04	22.109	22.934	22.872	22.876	24.941
0.05	21.276	21.930	21.986	21.986	24.231

Figure 2 shows the visual results in the frame #100 of video sequence "Flowers" by use in a) *FMRSTF* filter, in b) *FVMRSTF* filter, and in c) the proposed *FDARTF* filter. These images were recovered from Gaussian noise corrupted with variance 0.01, 0.015, 0.02, and 0.03 from top to bottom of Figure 2. From this Figure, one can see that the restored frames by means of use the proposed filters appear to have a better subjective quality.

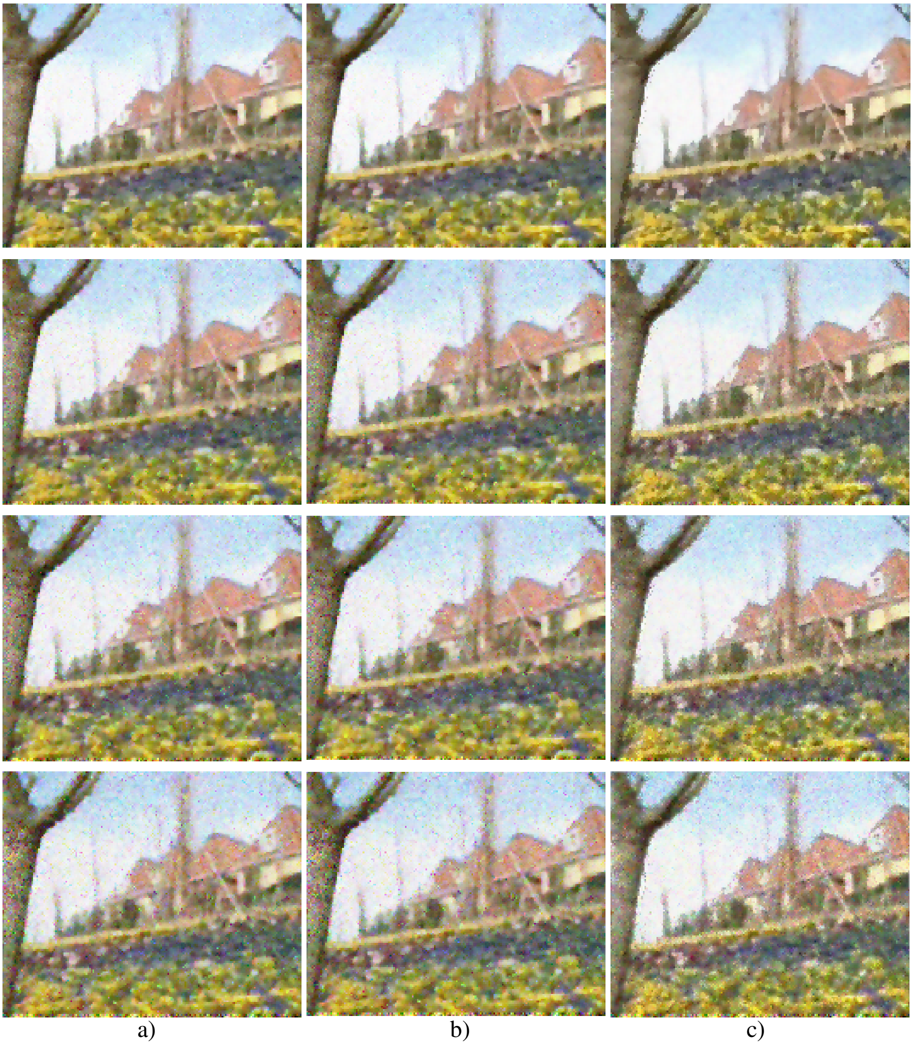

Fig. 2. Visual results in the frame #100 of video sequence "Flowers", a) Column of restored images by *FMRSTF*, b) Column of restored images by *FVMRSTF*, and c) Column of restored images by *FDARTF*. These images were recovered from Gaussian noise corrupted mages with variance 0.01, 0.015, 0.02, and 0.03 from top to bottom.

4 Conclusions

In this paper a novel robust adaptive recursive scheme for fuzzy logic based motion detection 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. It is demonstrated that taking into account, both robust characteristics (gradients and vectors) and connecting them together, we can realize a better algorithm, 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.

Acknowledgments. This work is supported by National Polytechnic Institute of Mexico and CONACyT.

References

1. Zlokolica, V., De Geyter, M., Schulte, S., Pizurica, A., Philips, W., Kerre, E.: Fuzzy Logic Recursive Motion Detection for Tracking and Denoising of Video Sequences. In: IS&T/SPIE Symposium on Electronic Imaging, San Jose, California, USA, pp. 771–782 (2005)
2. Zlokolica, V.: Advanced Non-Linear Methods for Video Denoising. PhD Thesis, Gent University (2006)
3. Zlokolica, V., Schulte, S., Pizurica, A., Philips, W., Kerre, E.: Fuzzy Logic Recursive Motion Detection and Denoising of Video Sequences. Journal of Electronic Imaging 15(2), 023008 (2006)
4. Trahanias, P.E., Venetsanopoulos, A.N.: Vector Directional Filters-A New Class of Multichannel Image Processing Filters. IEEE Transactions on Image Processing 2(4), 528–534 (1993)
5. Plataniotis, K.N., Venetsanopoulos, A.N.: Color Image Processing and Applications. Springer, Berlin (2000)
6. Schulte, S., De Witte, V., Nachtegael, M., Van der Weken, D., Kerre, E.: Fuzzy Two-Step Filter for Impulse Noise Reduction from Color Images. IEEE Transactions on Image Processing 15(11), 3567–3578 (2006)
7. Fuzzy Logic Fundamentals, ch. 3, pp. 61–103 (2001) www.informit.com/content/images/ 0135705991/ samplechapter/0135705991.pdf
8. Gallegos-Funes, F., Ponomaryov, V., De-La- Rosa, J.: ABST M-type K-nearest neighbor (ABSTM-KNN) for image denoising. IEICE Trans. Funds. Electronics Comms. Computer Science E88A(3), 798–799 (2005)
9. Ponomaryov, V.I., Gallegos-Funes, F.J., Rosales-Silva, A.: Real-Time Color Imaging Based on RM-Filters for Impulsive Noise Reduction. J. Imaging Science and Technology 49(3), 205–219 (2005)
10. Trahanias, P.E., Karakos, D.G., Venetsanopoulos, A.N.: Directional Processing of Color Images: Theory and Experimental Results. IEEE Trans. Image Process. 5(6), 868–880 (1996)
11. Plataniotis, K.N., Androutsos, D., Vinayagamoorthy, S., Venetsanopoulos, A.N.: Color Image Processing Using Adaptive Multichannel Filters. IEEE Trans. Image Process. 6(7), 933–949 (1997)