

Document downloaded from:

<http://hdl.handle.net/10251/38128>

This paper must be cited as:

Camarena Estruch, JG.; Gregori Gregori, V.; Morillas, S.; Sapena Piera, A. (2013). A simple fuzzy method to remove mixed Gaussian-Impulsive noise from color images. IEEE Transactions on Fuzzy Systems. 21(5):971-978. doi:10.1109/TFUZZ.2012.2234754.



The final publication is available at

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6384731>

Copyright Institute of Electrical and Electronics Engineers (IEEE)

A simple fuzzy method to remove mixed Gaussian-impulsive noise from colour images

Joan-Gerard Camarena, Valentín Gregori, Samuel Morillas and Almanzor Sapena

Abstract—Mixed impulsive and Gaussian noise reduction from digital colour images is a challenging task because it is necessary to appropriately process both types of noise that in turn need to be distinguished from the original image structures such as edges and details. Fuzzy theory is useful to build simple, efficient and effective solutions for this problem. In this paper, we propose a fuzzy method to reduce Gaussian and impulsive noise from colour images. Our method uses one only filtering operation: a weighted averaging. A fuzzy rule system is used to assign the weights in the averaging so that both noise types are reduced and image structures are preserved. We provide experimental results to show that the performance of the method is competitive with respect to state-of-the-art filters.

Index Terms—Color Image Filter, Fuzzy Metrics, Fuzzy Rules, Vector Median Filter

I. INTRODUCTION

SEVERAL sources and types of noise have been studied regarding digital colour images. In many cases, more than one noise type can contaminate the images. This happens for instance in colour images containing some Gaussian noise from the image acquisition phase (camera optics and CCD) that are also contaminated with impulse noise due to transmission errors or storage faults. Despite the existence of many filtering solutions to reduce the different noise types separately [1]-[3], only a few methods to process mixed noise have been published and, moreover, most of them are developed for gray-scale images.

The simplest way to reduce mixed noise from a digital image is to consecutively apply several (usually two) specific methods, one for each kind of noise in the image. However, the application of several filters could dramatically decrease the computational efficiency of the whole process which implies that this solution could not be practical for real applications. Therefore, it is more interesting to devise specific filters to remove mixed noise.

To date, a few methods in the literature are able to approach this problem efficiently. The *Peer Group Averaging* (PGA) technique presented in [4]-[7] and extended to the fuzzy context in [8] removes mixed noise by combining a statistical method for impulse noise detection and replacement with an averaging operation between the (fuzzy) peer group members to smooth out Gaussian noise. The difference between these methods relays on how to build the peer groups: [4], [5], [7] use the Fisher Linear Discriminant, [6] uses region analysis

and [8] uses fuzzy rules. The *Trilateral Filter* (TF) [9] is based on the well-known *Bilateral Filter* [10], [11] to smooth Gaussian noise but including an impulse detector to be also able to reject impulse noise. On the same basis, [12] includes a switching mechanism in the bilateral filter to remove impulses. The *Adaptive Nearest Neighbor Filter* (ANNF) and its variants [13], [14] use a weighted averaging where the weights are computed according to robust measures so that impulses that receive lower weights are reduced. The *Fuzzy Vector Median Filter* (FVMF) [15] performs a weighted averaging where the weight of each pixel is computed according to its similarity to the robust vector median. Another important family of filters are the partition based filters [16], [17] that classify each pixel to be processed into several signal activity categories which, in turn, are associated to appropriate processing methods. Similarly, [18] employs a Bayesian classification combined with kernel regression. Another well studied solution is the regularization approach [19]-[27] based on the minimization of appropriate energy functions by means of Partial Differential Equations (PDEs). In this context, recently, it has been proposed to use different minimization terms to reduce the different noise types [25], to employ segmentation followed by regularization [26], and to combine statistical noise detection and regularization [27]. Other approaches propose to employ image decomposition to separate and reduce noise [28], robust gradient vector flow and diffusion models [29], or finite element techniques [30]. Also, the problem of mixed Poisson-Gaussian noise removal is studied in [31].

In this paper we propose a simple method to remove mixed impulse and Gaussian noise from colour images which is based on fuzzy logic. According to our proposal, each image pixel is filtered only once using the same operation: a simple weighted average over the pixels in a filtering window. The adaptive nature of the method relays on how the weights involved are computed, for which we use a fuzzy-rule based system. This fuzzy system takes as input two sources of information on the pixels in the filtering window: (i) the degree of noisiness (from the impulsive point of view) computed using a statistical method, and (ii) the degrees of similarity between the central pixel and the rest of the pixels in the window. From this information, the proposed method computes the weights that allow to process each pixel in an appropriate way, reducing the noise and preserving the image structures appropriately.

The paper is organized as follows. Section II describes the proposed fuzzy method. Experimental results are provided in Section III and, finally, conclusions are drawn in Section IV.

The authors are members of the Instituto Universitario de Matemática Pura y Aplicada, Universidad Politécnica de Valencia, Spain. The authors acknowledge the support of Spanish Ministry of Science and Innovation under Grant MTM2009-12872-C02-01. Corresponding author's email: alsapie@mat.upv.es

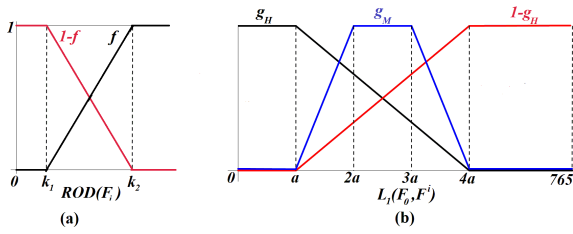


Fig. 1. (a) Noise degree of a pixel \mathbf{F}_i as a function f of $ROD(\mathbf{F}_i)$; (b) Similarity of \mathbf{F}_i with respect to \mathbf{F}_0 as a function of $L_1(\mathbf{F}_0, \mathbf{F}^i)$

II. PROPOSED FUZZY METHOD

Let \mathcal{F} be a colour image to be processed, and let W be a sliding filtering window, of size $n \times n$ ($n = 3, 5, \dots$), centered at the pixel \mathbf{F}_0 under processing. The vectors in \mathcal{F} are denoted as $\mathbf{F}_i = (F_i^R, F_i^G, F_i^B)$, as usual in the RGB colour space.

The proposed method, that we name *Simple Fuzzy Rule Filter* (SFRF), consists of replacing a pixel \mathbf{F}_0 of the image by a pixel $\hat{\mathbf{F}}_0$ which is a weighted average on certain selected pixels in W , denoted by $\mathbf{F}^0, \mathbf{F}^1, \dots, \mathbf{F}^m$, and it is given by

$$\hat{\mathbf{F}}_0 = \left(\sum_{i=0}^m w_i \cdot \mathbf{F}^i \right) / \left(\sum_{i=0}^m w_i \right), \quad (1)$$

where the weights $w_i \in [0, 1]$ are obtained through defuzzification using fuzzy logic inference by a fuzzy system. The adaptivity of the method is given by the usage of these weights and the way that they are computed. This set of weights is different for each filtering window and depends on the local features observed, which allows to appropriately process both types of noisy pixels as well as the original image structures.

In the following subsections, we detail how to obtain these weights for each filtering window. Mainly, we use two sources of information: First, the noisiness, from the impulsive point of view, of each pixel in the image; second, the similarities observed between the pixel under processing and the rest of the pixels in the window. From this information, a fuzzy rule based system obtains the weights through fuzzy inference.

A. Noisiness of image pixels

In the first step we evaluate how noisy each image pixel is. So, we assign a certainty degree $\delta(\mathbf{F}_i)$ for the vague statement “ \mathbf{F}_i is noisy” to each \mathbf{F}_i as follows.

We order the pixels \mathbf{F}_j in a window W' centered at \mathbf{F}_i which is also taken, for simplicity, of size $n \times n$ in the way $\mathbf{F}_{(0)}, \mathbf{F}_{(1)}, \dots, \mathbf{F}_{(n^2-1)}$ according to a distance measure ρ , so that $\rho(\mathbf{F}_i, \mathbf{F}_{(0)}) \leq \rho(\mathbf{F}_i, \mathbf{F}_{(1)}) \leq \dots \leq \rho(\mathbf{F}_i, \mathbf{F}_{(n^2-1)})$, where obviously $\mathbf{F}_{(0)} = \mathbf{F}_i$. As the distance measure ρ we use the metric L_∞ given by

$$L_\infty(\mathbf{F}_i, \mathbf{F}_j) = \max\{|F_i^R - F_j^R|, |F_i^G - F_j^G|, |F_i^B - F_j^B|\}, \quad (2)$$

for its high sensitivity to impulse noise detection (see [32]).

Now, we consider the $s + 1$ first pixels $\mathbf{F}_{(0)}, \mathbf{F}_{(1)}, \dots, \mathbf{F}_{(s)}$ and compute the ROD_s statistic for the pixel \mathbf{F}_i , which is an adaptation of the definition in [9] to the colour case, as:

$$ROD_s(\mathbf{F}_i) = \sum_{j=0}^s L_\infty(\mathbf{F}_i, \mathbf{F}_{(j)}) \quad (3)$$

Since $\mathbf{F}_i = \mathbf{F}_{(0)}$ then $L_\infty(\mathbf{F}_i, \mathbf{F}_{(0)}) = 0$ and ROD_s takes integer values in the interval $[0, 255s]$. A low value of

$ROD_s(\mathbf{F}_i)$ means that the selected $s + 1$ pixels $\mathbf{F}_{(j)}$ in W' are close to \mathbf{F}_i which in turn means that \mathbf{F}_i is expected to be noise-free. Moreover, higher values of $ROD_s(\mathbf{F}_i)$ indicate a higher noise degree for \mathbf{F}_i , since no close pixels are found. The setting of the parameter s is discussed in Section III.

Now, if we put $x = ROD_s(\mathbf{F}_i)$ we define the certainty degree $\delta(\mathbf{F}_i)$ for the vague statement “ \mathbf{F}_i is noisy” by

$$\delta(\mathbf{F}_i) = f(x) = \begin{cases} 0 & x \leq k_1 \\ \frac{x - k_1}{k_2 - k_1} & k_1 < x < k_2 \\ 1 & k_2 \leq x \end{cases}, \quad (4)$$

where the settings of the parameters k_1 and k_2 will be commented in Section III. Notice that in all this work we are employing linear membership functions instead of exponential membership functions to lower the computational complexity.

Finally, we assign to each pixel \mathbf{F}_i of \mathcal{F} a certainty degree of the vague statement “ \mathbf{F}_i is not noisy”. Representing the negation by the fuzzy involutive operator, it will be given by $1 - \delta(\mathbf{F}_i)$. The corresponding fuzzy sets, f and $1 - f$, defined on $[0, 255s]$ are shown in Figure 1 (a).

B. Similarities between the pixel under processing and the rest of the pixels in the window

In the second step we are interested in analyzing the similarity between the pixel under processing \mathbf{F}_0 and the rest of the pixels in the sliding window W . To measure the similarity between two pixels we now use the metric L_1 :

$$L_1(\mathbf{F}_i, \mathbf{F}_j) = |F_i^R - F_j^R| + |F_i^G - F_j^G| + |F_i^B - F_j^B| \quad (5)$$

Here, we prefer to use this metric instead of the L_∞ above because the latter is not suitable to measure similarity since it focus on the biggest difference found among the colour components and not in their similarities. The L_1 metric is much better for measuring the similarity because its computation takes into account the three components of the two pixels under comparison and the differences between the pixels components are not emphasized, unlike other metrics such as the L_∞ or the Euclidean (L_2), where they are powered. In fact, in [32], [33], [34] it has been studied the influence on a filter performance of the metric used and it was concluded that depending on the purpose some metrics are more suitable than others. Therefore, here we choose using two different metrics in two different parts of the filter because, as we have seen, the purpose of each part is totally different. On the other hand, the additional computational cost is practically negligible because the absolute value differences have been evaluated before, when computing L_∞ .

Now, using the similarities observed, we will assign to certain selected pixels of W , denoted by \mathbf{F}^i , a certainty degree in the vague statements: *the similarity between \mathbf{F}^i and \mathbf{F}_0 is “high”, “medium” and “low”,* denoted by $\mu_H(\mathbf{F}_0, \mathbf{F}^i)$, $\mu_M(\mathbf{F}_0, \mathbf{F}^i)$ and $\mu_L(\mathbf{F}_0, \mathbf{F}^i)$, respectively.

But previously, we select the pixels that will be involved in the filtering operation. Using the distance $L_1(\mathbf{F}_0, \mathbf{F}_j)$ between each pixel \mathbf{F}_j of W and the pixel under processing \mathbf{F}_0 we introduce a new ordering for the n^2 pixels of W in the ordered set $W^* = \{\mathbf{F}^0, \mathbf{F}^1, \dots, \mathbf{F}^{n^2-1}\}$ such that

$$L_1(\mathbf{F}_0, \mathbf{F}^0) \leq L_1(\mathbf{F}_0, \mathbf{F}^1) \leq \dots \leq L_1(\mathbf{F}_0, \mathbf{F}^{n^2-1})$$

where, obviously, $\mathbf{F}^0 = \mathbf{F}_0$. Then, we select the first $m+1$ pixels $\mathbf{F}^0, \dots, \mathbf{F}^m$ to avoid pixels very different from \mathbf{F}_0 being involved in the filtering. The setting of the parameter m is discussed in Section III.

Now, to assign the certainty degrees of the three vague propositions above we perform as follows: we put $x = L_1(\mathbf{F}_0, \mathbf{F}^i)$ and we define $\mu_H(\mathbf{F}_0, \mathbf{F}^i) = g_H(x)$ by

$$g_H(x) = \begin{cases} 1 & x \leq a \\ -x/(3a) + 4/3 & a < x < 4a \\ 0 & 4a \leq x \end{cases} \quad (6)$$

Using the fuzzy negation, we assign $\mu_L(\mathbf{F}_0, \mathbf{F}^i) = 1 - \mu_H(\mathbf{F}_0, \mathbf{F}^i)$. The certainty for $\mu_M(\mathbf{F}_0, \mathbf{F}^i) = g_M(x)$ is

$$g_M(x) = \begin{cases} (x-a)/a & a < x < 2a \\ 1 & 2a \leq x \leq 3a \\ (4a-x)/a & 3a < x < 4a \\ 0 & \text{elsewhere} \end{cases} \quad (7)$$

The corresponding fuzzy sets, g_H , $1-g_H$, and g_M defined on $[0, 3 \cdot 255]$ are represented in Figure 1 (b). The best value of the a parameter in (6) and (7) depends on the noise intensity, and its optimization will be discussed in Section III.

C. Fuzzy system and computation of weights

To compute the weights involved in the filtering average operation we now use a fuzzy rule based system and fuzzy inference. The fuzzy system use the vague statements described in the previous subsections to decide whether each weight in (1) should be *large*, *medium* or *small*. Finally, defuzzification is used to obtain the particular value for each weight.

The objective of the rules in the fuzzy system can be summarized in two main ideas: (i) pixels that are noisy should be assigned to a small weight, and (ii) pixels that are noise-free can only be associated to a larger weight if either they are similar to the central pixel or if the central pixel is noisy. This latter idea, for the different cases to be found, is summarized in the following tree fuzzy rules:

- 1) IF (\mathbf{F}^i is not noisy AND \mathbf{F}_0 is noisy AND the similarity between \mathbf{F}_0 and \mathbf{F}^i is medium)
THEN w_i is a medium weight
- 2) IF (\mathbf{F}^i is not noisy AND \mathbf{F}_0 is noisy AND the similarity between \mathbf{F}^i and \mathbf{F}_0 is low)
OR (\mathbf{F}^i is not noisy AND \mathbf{F}_0 is not noisy AND the similarity between \mathbf{F}^i and \mathbf{F}_0 is high)
THEN w_i is a large weight
- 3) IF (\mathbf{F}^i is noisy)
OR (\mathbf{F}^i is not noisy AND \mathbf{F}_0 is noisy AND the similarity between \mathbf{F}^i and \mathbf{F}_0 is high)
OR (\mathbf{F}^i is not noisy AND \mathbf{F}_0 is not noisy AND the similarity between \mathbf{F}^i and \mathbf{F}_0 is medium)
OR (\mathbf{F}^i is not noisy AND \mathbf{F}_0 is not noisy AND the similarity between \mathbf{F}^i and \mathbf{F}_0 is low)
THEN w_i is a small weight

Before performing the fuzzy inference process we need to define the fuzzy sets corresponding to the consequents of the fuzzy rules. Each weight $w_i \in [0, 1]$ is associated to a certainty

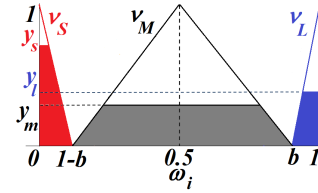


Fig. 2. Fuzzy sets ν_L , ν_M and ν_S .

degree in the vague statements “ w_i is a large weight”, “ w_i is a medium weight” and “ w_i is a small weight” denoted by $\nu_L(w_i)$, $\nu_M(w_i)$ and $\nu_S(w_i)$, respectively. The fuzzy sets ν_L , ν_M and ν_S are represented in Figure 2, where we have selected triangular-shape fuzzy membership functions for simplicity of the defuzzification process, as follows:

$$\nu_M(w_i) = \begin{cases} (2w_i - 1)/(2b - 1) + 1 & 1 - b < w_i \leq 0.5 \\ (1 - 2w_i)/(2b - 1) + 1 & 0.5 < w_i < b \\ 0 & \text{elsewhere} \end{cases} \quad (8)$$

$$\nu_L(w_i) = (w_i - 1)/(1 - b) + 1 \text{ if } b < w_i \leq 1 \text{ and } 0 \text{ elsewhere} \quad (9)$$

$$\nu_S(w_i) = w_i/(b - 1) + 1 \text{ if } 0 \leq w_i \leq 1 - b \text{ and } 0 \text{ elsewhere} \quad (10)$$

The value of the b parameter has been set experimentally and it will be discussed in Section III.

Now, for computing the certainty degree of the antecedents of the Fuzzy Rules, following the usual procedure in fuzzy logic, we apply the conjunction operation **AND** and the disjunction operation **OR** by means of a t-norm $*$ and its associated s-norm $'$, respectively. In this paper we use the usual product as the t-norm and the probabilistic addition as the s-norm.

The certainties of the antecedents are assigned to the consequents, and finally, by defuzzification, we obtain the weight w_i of the pixel \mathbf{F}^i . In so doing, we have used the centroid technique, or center of gravity (**COG**), which is the most popular defuzzification method ([35], [36], [37]) as follows.

Suppose that, for a pixel \mathbf{F}^i , y_m , y_l and y_s are the certainty degrees of the consequents in rules 1), 2) and 3) above, respectively. In Figure 2, consider the triangles defined by ν_M , ν_L and ν_S and the constant functions $y = y_m$, $y = y_l$, $y = y_s$. Taking the surface in each of the mentioned triangles and under each one of the three constant functions, respectively, three trapeziums are built. The polygonal line constituted by the tops and sides of these trapeziums determines a fuzzy set A on $[0, 1]$ which is integrable in the Riemann (classical) sense. The abscissa of the center of gravity of the area under A is the weight w_i . So, $w_i = \left(\int_0^1 x \cdot A(x) dx \right) / \left(\int_0^1 A(x) dx \right)$

The weights for the pixels \mathbf{F}^i for $i = 0, 1, \dots, m$ allow to apply equation 1 for obtaining the desired denoised pixel $\hat{\mathbf{F}}_0$.

III. EXPERIMENTAL STUDY

In this section, several images have been considered to evaluate the performance of the proposed filter. Two types of impulse noise (fixed-value and random-value) and the classical model for Gaussian noise [1], [38] have been considered.

The images of Lenna, Flower, Parrots and Motorbikes (see Fig. 3) have been corrupted with Gaussian noise followed by

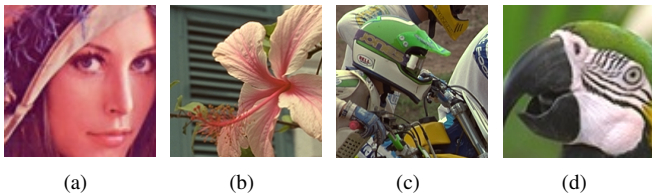


Fig. 3. Test Images: (a) Lenna, (b) Flower [39], (c) Motorbikes [39], (d) Parrots

TABLE I
PERFORMANCE IN TERMS OF AVERAGE PSNR WHEN FILTERING THE IMAGES FLOWER AND LENA CONTAMINATED WITH DIFFERENT DENSITIES OF THE TWO TYPES OF MIXED NOISE CONSIDERED FOR DIFFERENT VALUES OF THE s PARAMETER.

Noise	$s = 2$	$s = 3$	$s = 4$	$s = 5$
$\sigma = 5, p = 0.05$	31.75	30.78	31.05	30.19
$\sigma = 10, p = 0.1$	29.66	29.63	29.62	28.83
$\sigma = 15, p = 0.15$	27.77	27.72	27.19	26.80
$\sigma = 20, p = 0.2$	26.24	26.23	26.32	25.55
$\sigma = 25, p = 0.25$	25.64	25.47	25.05	24.48
$\sigma = 30, p = 0.3$	24.92	24.49	24.45	24.69

one type of impulse noise with different noise intensities. From now on, we represent the standard deviation of Gaussian noise as $\sigma \in [5, 30]$ and the probability of impulse noise appearance as $p \in [0.05, 0.3]$.

The filters performance has been evaluated by using the common objective measures MAE (*Mean Absolute Error*), that measures the detail preserving, PSNR (*Peak Signal to Noise Ratio*), that measures the noise suppression, and NCD (*Normalized Color Difference*), that measures the color preservation, as defined in [1], [38].

A. Adjustment of Parameters

In the proposed filter, several parameters are involved (m, s, k_1, k_2, a, b). For their adjustment, we have analyzed the PSNR performance as a function of them and for it we have used the test images of Flowers and Lenna (Figure 3 (e), (f)), corrupted with the two mixtures of the different noise models (Gaussian and random-value impulse and Gaussian and fixed-value impulse).

First, recall that the value m is the number of weighted pixels in the filtering window which will compute in the output (1). The value of the parameter s indicates the number of pixels which will compute in Eq. (3) to determine the noisy degree of a pixel. Both parameters have been selected according to the window size taking into account that it should involve a minimum number of pixels to be able to suppress the Gaussian noise and less than a maximum number of pixels to avoid excessive image blurring. We have extensively studied the 3×3 case, which is commonly used in the literature [8], and from our experiences the most appropriate setting to optimize overall PSNR is $m = 7$ and $s = 2$. Table I shows the average PSNR for the images Flower and Lenna contaminated with different densities of the two types of mixed noise for different values of s , where we can see that $s = 2$ provides the best overall results but that a higher value of s ($s = 3, 4$) could be used without significantly decreasing the performance. An analogous procedure has led us to set $m = 7$.

The k_1 and k_2 parameters in (4) are used to determine the noise degree of a pixel from $ROD_s(\mathbf{F}_i)$. Recall that if the value of $ROD_s(\mathbf{F}_i)$ is lower than k_1 then the pixel has a noise degree of 0, whereas if $ROD_s(\mathbf{F}_i)$ is greater than k_2 then the pixel has a noise degree of 1. For intermediate values the corresponding certainty is given by a linear ascending relation, as explained before. The setting of k_1 and k_2 is very important to obtain appropriate certainty degrees and should be set according to the image noise level. Our experiences have determined that PSNR optimal values for k_1 and k_2 are obtained for $k_1 \in [0.45ROD_{max}, 0.55ROD_{max}]$ and $k_2 \in [0.55ROD_{max}, 0.65ROD_{max}]$ where $ROD_{max} = \max\{ROD_s(\mathbf{F}_i) : \mathbf{F}_i \in \mathcal{F}\}$. Therefore, these parameters can be automatically set in the filter. In particular, we have fixed the values $k_1 = 0.5ROD_{max}$ and $k_2 = 0.6ROD_{max}$ as robust settings for the proposed filter.

The a parameter in Eq. (6) and Eq. (7) determines how the degrees of similarity are computed. For lower noise levels, a small value of a is appropriate but the value should increase as the noise level grows. We have tried to relate the appropriate value for a with the density of Gaussian noise in the image, since it critically influences the observed similarities. We have obtained the PSNR optimal values for a varying proportionally p and σ in their corresponding ranges. Then, we performed a linear regression study, that relates a with σ as $a = 0.998\sigma + 1.960$. So, we have obtained a way to automatically set a adaptively to the noise level. Notice that this noise level can be estimated using the technique described in [8].

Finally, the b parameter in Eqs. (8-10) is used to determine the weight associated to each pixel in the image through defuzzification of the corresponding certainty degrees of the vague statements. After several experiences, our results show that an appropriate setting is $b = 0.9$.

B. Performance of the proposed method.

First we assess the performance of the proposed *Simple Fuzzy Rule Filter* (SFRF) in the noise detection phase. We assume that all pixels are contaminated with Gaussian noise so we just evaluate the performance of the impulse noise detection. Given that our noise detection is fuzzy, we compute as a measure of the detection error the absolute value difference between the fuzzy noise degree $\delta(\mathbf{F}_i)$, determined for each pixel \mathbf{F}_i , and the values 1 or 0 depending on whether the original pixel has been corrupted with impulse noise or not. Since we apply the filtering process iteratively, after k iterations we have k different noise degrees $\delta(\mathbf{F}_i)^1, \delta(\mathbf{F}_i)^2, \dots, \delta(\mathbf{F}_i)^k$. We take as global noise degree $\delta(\mathbf{F}_i)^G$ to measure the detection error the result of applying the **OR** operation among all these k degrees, that is: $\delta(\mathbf{F}_i)^G = \delta(\mathbf{F}_i)^1 \text{ OR } \delta(\mathbf{F}_i)^2 \text{ OR } \dots \text{ OR } \delta(\mathbf{F}_i)^k$, using the probabilistic addition as the s-norm modeling the **OR** operation. The detection performance for a whole image is given by the average of all absolute value differences computed. The results shown in Table II suggest that the impulse noise detection is accurate, since detection errors obtained are low. In addition, it should be taken into account that the images are also contaminated with Gaussian noise, which means that when a pixel is contaminated only

TABLE II

ERROR IN IMPULSE NOISE DETECTION MEASURED IN TERMS OF MEAN ABSOLUTE ERROR OVER THE FLOWER AND MOTORBIKES IMAGES CONTAMINATED WITH DIFFERENT DENSITIES OF MIXED GAUSSIAN AND FIXED/RANDOM-VALUE IMPULSE NOISE, CORRECTION TERM c AND CORRECTED ERRORS (INDICATED WITH *).

Image	$\sigma = 5$	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$	$\sigma = 30$
	and $p = 0.05$	and $p = 0.1$	and $p = 0.15$	and $p = 0.2$	and $p = 0.25$	and $p = 0.3$
Flower Random-value	0.040	0.054	0.109	0.136	0.135	0.175
Flower Fixed-value	0.056	0.052	0.076	0.104	0.117	0.135
Motorbikes Random-value	0.016	0.018	0.035	0.053	0.075	0.098
Motorbikes Fixed-value	0.024	0.015	0.028	0.037	0.075	0.114
Correction term c	0.000	0.000	0.000	0.001	0.008	0.037
Flower Random-value*	0.040	0.054	0.109	0.135	0.127	0.138
Flower Fixed-value*	0.056	0.052	0.076	0.103	0.109	0.098
Motorbikes Random-value*	0.016	0.018	0.035	0.052	0.067	0.061
Motorbikes Fixed-value*	0.024	0.015	0.028	0.036	0.067	0.077

TABLE III

COMPARISON OF THE PERFORMANCE MEASURED IN TERMS OF MAE, PSNR, AND NCD ($\times 10^2$) USING THE TEST IMAGES CONTAMINATED WITH DIFFERENT DENSITIES OF MIXED GAUSSIAN AND RANDOM-VALUE IMPULSIVE NOISE

Filter	$\sigma = 5$ Gaussian and $p = 0.05$ impulse			$\sigma = 10$ Gaussian and $p = 0.1$ impulse			$\sigma = 20$ Gaussian and $p = 0.2$ impulse			$\sigma = 30$ Gaussian and $p = 0.3$ impulse		
	MAE	PSNR	NCD	MAE	PSNR	NCD	MAE	PSNR	NCD	MAE	PSNR	NCD
Parriss												
None	7.87	20.54	8.23	14.06	18.13	14.77	27.31	14.71	28.43	39.04	12.62	39.76
VMF	12.63	20.18	6.36	13.71	19.97	7.76	17.06	19.06	10.74	21.08	17.98	14.38
AMF	13.96	20.93	7.72	16.04	20.35	9.81	21.40	18.70	14.64	27.37	17.00	18.67
ANNF	12.89	20.3	5.69	13.28	20.22	6.25	15.14	19.86	8.12	18.41	18.98	10.48
PGA	9.82	22.91	5.53	11.02	22.38	7.08	14.28	20.55	9.72	18.04	18.86	11.89
TF	6.34	24.66	5.66	10.74	21.67	7.55	14.79	19.22	10.63	17.71	18.27	11.96
TF 5×5	6.49	23.01	6.92	9.75	21.43	9.12	15.97	18.67	14.44	21.66	17.24	18.66
FVMF	12.41	20.22	6.12	13.16	20.04	7.00	15.46	19.33	8.76	18.04	18.54	10.68
FPGA	5.95	26.43	3.64	9.26	22.52	5.49	12.80	20.42	7.82	15.03	19.85	9.48
SFRF	6.33	26.32	3.72	7.38	25.66	4.71	11.18	22.52	7.06	15.30	20.43	9.64
SFRF 5×5	5.97	25.28	3.71	8.28	23.57	5.74	14.95	19.82	10.27	18.89	18.45	12.47
SFRF 7×7	5.49	26.28	3.74	7.61	24.99	5.60	13.50	21.23	10.09	19.06	18.78	13.38
Lenna												
None	7.88	20.79	8.24	14.27	18.26	15.23	27.68	14.76	28.24	37.43	13.17	38.40
VMF	6.77	27.02	5.01	8.31	25.91	6.55	11.64	23.68	9.90	15.24	21.84	13.52
AMF	8.99	25.59	7.10	11.09	24.09	9.85	16.74	21.06	15.30	21.29	19.23	19.29
ANNF	6.81	26.99	4.41	7.42	26.63	5.21	9.38	25.38	7.45	12.29	23.60	10.04
PGA	5.92	28.58	4.49	7.44	27.30	6.23	10.11	24.80	8.71	12.73	23.07	10.73
TF	4.71	27.12	5.08	7.14	26.15	6.31	9.70	24.44	8.12	12.12	23.23	10.32
TF 5×5	6.13	24.06	6.57	8.39	23.02	8.59	13.55	20.34	13.11	16.59	19.92	15.63
FVMF	6.59	27.05	4.81	7.77	26.09	5.80	9.68	24.81	7.88	11.93	23.47	9.54
FPGA	4.10	31.13	3.21	6.02	28.23	4.69	8.24	26.11	6.77	10.56	24.56	9.00
SFRF	3.86	32.65	3.00	5.42	29.28	4.83	8.30	26.17	6.83	11.78	23.49	9.85
SFRF 5×5	3.64	32.35	3.00	5.64	29.30	5.05	9.95	25.28	8.91	12.84	23.28	10.96
SFRF 7×7	4.05	30.53	3.28	6.20	27.73	5.30	10.93	24.07	9.57	14.66	22.12	12.92
Motorbikes												
None	7.71	21.09	9.77	14.74	18.11	18.77	27.87	14.82	34.20	39.02	12.92	46.34
VMF	8.74	24.41	6.77	10.62	23.43	8.97	14.00	21.90	13.18	17.95	20.19	17.69
AMF	10.63	24.07	8.61	13.15	22.72	11.87	18.26	20.37	17.15	25.42	18.46	21.51
ANNF	8.81	24.63	5.88	9.58	24.28	6.96	11.66	23.30	9.66	14.94	21.62	12.77
PGA	7.26	26.70	6.15	8.81	25.51	8.36	11.84	23.18	11.46	15.05	21.35	14.19
TF	5.283	26.32	6.71	8.55	24.47	8.63	12.00	22.41	11.33	14.79	21.24	13.83
TF 5×5	6.10	23.87	8.08	9.24	22.15	11.12	14.6	19.94	16.24	18.54	18.91	19.92
FVMF	8.60	24.39	6.62	9.73	23.82	7.93	12.12	22.55	10.32	14.71	21.29	13.12
FPGA	4.63	29.94	4.29	7.11	26.19	6.32	10.01	23.93	9.00	12.51	22.66	11.82
SFRF	4.65	30.49	4.18	6.63	27.79	6.06	10.23	24.19	9.03	14.15	21.66	12.39
SFRF 5×5	4.23	30.01	4.08	6.47	27.39	6.78	11.40	23.64	11.86	15.05	21.62	14.76
SFRF 7×7	4.44	29.01	4.34	6.89	26.37	6.98	12.16	22.82	12.57	16.50	20.77	16.17
Flower												
None	7.69	21.25	8.23	14.50	18.39	15.68	27.10	15.16	29.29	38.26	13.21	40.83
VMF	6.91	27.15	4.61	8.71	25.80	6.39	12.04	23.62	9.71	15.55	21.66	13.29
AMF	8.81	25.79	6.28	11.20	24.15	8.93	15.84	21.54	13.34	20.73	19.47	16.93
ANNF	7.17	27.01	3.90	7.82	26.61	4.71	9.67	25.32	6.98	12.46	23.46	9.14
PGA	6.17	28.40	4.21	7.76	27.08	6.06	10.72	24.52	8.94	13.21	22.92	10.76
TF	4.66	27.89	5.05	7.55	26.53	6.02	10.41	24.43	7.78	12.39	23.38	10.43
TF 5×5	5.95	24.56	6.58	8.65	23.40	8.74	13.04	21.32	12.52	16.47	20.29	15.60
FVMF	6.79	27.15	4.49	8.22	26.05	5.66	10.41	24.57	7.75	12.61	23.20	9.60
FPGA	4.45	31.04	3.06	6.39	28.29	4.52	8.79	25.96	6.93	10.96	24.48	8.71
SFRF	3.83	32.44	2.92	5.88	29.45	4.33	9.26	25.55	6.91	11.93	23.56	8.79
SFRF 5×5	3.76	32.20	2.92	5.99	29.10	5.11	10.37	25.10	9.39	13.19	23.20	11.17
SFRF 7×7	4.09	30.65	3.12	6.48	27.64	5.32	11.39	23.97	10.03	15.05	22.01	12.76

with a high Gaussian noise, it could also be considered an impulse. This is appropriate from the noise reduction point of view but it is considered an error by our statistic. To take this into account we introduce a correction term to our statistic. The correction term c represents the probability of an image pixel to be highly contaminated with Gaussian noise so that our method could detect it as an impulse. We compute c , which depends on σ , as the probability of any of the Gaussian random values added to the R , G , and B channels to be higher than th or lower than $-th$ setting $th = 75$. The corrected mean error is obtained by subtracting c from the original error. Notice that a correction term c significantly larger than 0 can only be found for high density Gaussian noise.

Second, the performance of the SFRF is compared with the classical *Arithmetic Mean Filter* (AMF), the *Vector Median*

Filter (VMF) and a series of state-of-the-art filters able to reduce both Gaussian and impulse noise: *Adaptive Nearest Neighbor Filter* (ANNF) [13], [14], *Trilateral Filter* (TF) [9](applied in a componentwise manner), *Fuzzy Vector Median Filter* (FVMF) [15], *Peer Group Averaging* (PGA) [7], and *Fuzzy Peer Group Averaging Filter* (FPGA) [8]. All filters have been applied on a 3×3 filter window in an iterative fashion with the same stop condition. Also, the proposed SFRF and the TF have been used with larger filtering windows and one only filtering (no iterations), adjusting the parameters appropriately.

From the experimental results in Tables III-IV it can be observed that for both types of mixed noise considered, the two best-performing filters are the proposed SFRF and the FPGA (the best results are marked in bold). The PGA and the TF are able to provide competitive performance only for

TABLE IV
COMPARISON OF THE PERFORMANCE MEASURED IN TERMS OF MAE, PSNR, AND NCD ($\times 10^2$) USING THE TEST IMAGES CONTAMINATED WITH DIFFERENT DENSITIES OF MIXED GAUSSIAN AND FIXED-VALUE IMPULSIVE NOISE

Filter	$\sigma = 5$ Gaussian and $p = 0.05$ impulse			$\sigma = 10$ Gaussian and $p = 0.1$ impulse			$\sigma = 20$ Gaussian and $p = 0.2$ impulse			$\sigma = 30$ Gaussian and $p = 0.3$ impulse		
	MAE	PSNR	NCD	MAE	PSNR	NCD	MAE	PSNR	NCD	MAE	PSNR	NCD
Parrtos												
None	6.11	21.73	8.08	11.84	18.84	15.70	23.69	15.02	30.23	32.96	13.38	41.92
VMF	12.17	20.50	6.20	13.38	20.14	7.70	16.10	19.27	10.91	18.32	18.65	14.00
AMF	13.08	21.29	7.24	14.61	20.78	9.43	18.35	19.57	13.67	21.04	18.73	16.96
ANNF	12.65	20.40	5.63	13.02	20.34	6.21	14.16	20.05	7.741	15.66	19.56	9.76
PGA	9.40	23.32	5.56	10.75	22.85	7.40	14.13	20.52	10.15	16.05	19.6	11.79
TF	5.48	26.46	5.00	10.31	22.10	7.41	14.28	19.18	9.96	16.69	18.16	12.58
TF 5×5	4.89	25.41	6.175	7.78	23.74	8.31	13.54	19.59	13.56	17.40	18.26	17.81
FVMF	12.00	20.40	6.04	13.02	20.06	7.03	14.54	19.52	8.33	16.12	18.74	9.54
FPGA	5.93	26.05	3.71	8.98	22.84	5.48	12.29	20.85	7.68	13.96	19.99	9.17
SFRF	5.89	27.52	3.61	7.11	26.20	4.73	10.00	23.38	6.76	12.41	21.88	8.79
SFRF 5×5	5.43	26.53	3.66	7.73	24.50	5.86	13.55	21.01	10.82	18.05	18.84	14.04
SFRF 7×7	4.69	29.32	3.61	6.77	27.14	5.67	11.89	22.79	10.69	16.55	20.41	15.37
Lenna												
None	6.07	22.31	7.88	12.10	18.80	15.12	23.45	15.43	28.24	34.10	13.28	39.76
VMF	6.55	27.28	4.86	8.11	26.09	6.42	11.10	24.13	9.83	13.36	22.87	12.47
AMF	7.74	26.55	6.12	9.70	25.06	8.36	13.41	22.84	12.52	17.05	20.99	16.45
ANNF	6.67	27.11	4.31	7.23	26.80	5.00	8.52	26.02	6.62	10.22	24.78	8.61
PGA	5.81	28.80	4.59	7.34	27.38	6.35	9.88	24.89	8.67	11.36	23.98	9.90
TF	3.82	30.32	4.45	6.67	27.20	5.80	8.88	25.25	7.86	11.24	23.76	9.54
TF 5×5	4.48	27.45	5.56	6.62	25.74	7.20	10.69	22.43	10.77	14.28	20.28	14.37
FVMF	6.40	27.30	4.67	7.48	26.42	5.56	9.16	25.17	7.44	10.68	24.33	8.97
FPGA	3.98	31.35	3.26	5.97	27.80	4.78	7.98	26.54	6.60	9.79	25.00	8.57
SFRF	3.44	33.52	2.94	5.07	30.64	4.07	7.78	26.66	6.24	9.61	25.19	8.42
SFRF 5×5	3.40	33.81	2.91	5.32	30.34	4.87	9.53	25.83	9.00	12.51	23.69	11.71
SFRF 7×7	3.54	33.43	3.06	5.34	29.94	5.04	10.01	25.12	9.57	14.01	22.60	13.46
Motorbikes												
None	6.20	22.08	9.91	12.36	18.74	19.33	24.21	15.22	36.14	35.01	13.17	50.30
VMF	8.55	24.59	6.69	10.15	23.79	8.76	13.02	22.47	12.96	16.03	21.11	17.76
AMF	9.78	24.57	8.13	11.65	23.54	11.22	15.37	21.69	16.39	18.74	20.19	20.40
ANNF	8.68	24.73	5.81	9.26	24.50	6.76	10.78	23.87	9.21	12.86	22.66	11.63
PGA	7.11	26.96	6.36	8.91	25.41	8.69	11.45	23.51	11.79	13.35	22.31	13.59
TF	4.48	28.31	6.23	7.97	25.41	8.08	11.24	22.95	10.80	13.61	21.69	13.88
TF 5×5	4.72	26.05	7.62	7.27	24.55	9.90	12.16	21.13	15.07	16.12	19.26	19.33
FVMF	8.24	24.72	6.45	9.21	24.26	7.67	11.14	23.20	9.79	12.91	22.29	12.21
FPGA	4.58	29.56	4.41	6.96	26.22	6.43	9.51	24.61	9.27	11.75	23.23	11.67
SFRF	4.05	31.67	4.15	6.24	28.33	6.03	8.94	25.53	8.89	11.81	23.23	11.62
SFRF 5×5	3.95	31.20	4.07	6.03	28.44	6.85	10.80	24.29	12.56	14.65	21.65	16.46
SFRF 7×7	3.93	31.52	4.26	6.09	28.26	7.04	10.94	23.97	13.25	15.93	21.11	19.11
Flower												
None	6.23	22.19	8.25	12.48	18.70	16.14	24.17	15.36	30.87	35.07	13.32	43.88
VMF	7.06	27.01	4.63	8.45	26.10	6.33	11.34	24.17	9.75	14.03	22.56	13.10
AMF	8.05	26.40	5.82	10.06	24.86	8.51	13.63	22.76	12.77	16.92	21.06	15.95
ANNF	7.06	27.19	3.83	7.63	26.82	4.63	9.09	25.91	6.85	11.09	24.41	8.76
PGA	6.02	28.78	4.33	7.59	27.31	6.26	10.34	24.85	9.11	12.12	23.64	10.62
TF	3.98	29.56	4.66	7.16	27.27	5.76	9.35	25.40	7.68	11.55	23.68	9.77
TF 5×5	4.64	26.76	5.99	7.18	25.09	7.94	11.41	22.14	11.97	14.99	20.14	15.59
FVMF	6.57	27.47	4.38	7.90	26.40	5.53	9.81	25.15	7.66	11.39	24.04	9.17
FPGA	4.46	30.30	3.21	6.36	27.66	4.67	8.54	26.37	7.05	10.77	24.59	8.99
SFRF	3.64	33.08	2.93	5.93	27.88	5.42	8.61	26.28	6.84	10.73	24.57	8.57
SFRF 5×5	3.54	33.33	2.91	5.65	30.01	5.16	10.10	22.53	10.29	13.51	23.11	13.20
SFRF 7×7	3.69	32.70	3.08	5.90	29.15	5.33	10.81	24.49	11.02	15.34	21.78	15.70

low noise. On the other hand, for higher noise, the ANNF and FVMF perform pretty robustly. The rest of the filters in the comparison (AMF, VMF) perform quite worse in general. The use of a larger filtering window instead of an iterative filtering seems to work well only in some cases and for low noise.

In Figure 4 we show some outputs of the best performing filters in each case, where we can see that the performance of the proposed method is also competitive from the visual point of view. For instance, in Figures 4 (a)-(h) we can see that SFRF performs better than FPGA and PGA, generating more smoothed flat regions and appropriately preserving image edges. However, in Figures 4 (i)-(p), SRFR is able to better reduce the noise than the PGA and ANNF but a little worse than FPGA. But, on the other hand, SFRF preserves better the image details. It should be pointed out that these filtering results use a parameter setting which has been obtained through PSNR optimization. On the one hand, it is known that PSNR does not perfectly match visual quality. Also, on the other hand since, in general, images contain more plain areas than edges, the former weigh more in the adjustment than the latter, which makes that optimal PSNR does not perfectly correspond with optimal visual image quality in the sense of edge preservation. To optimize visual quality, a different adjustment could be made based on subjective criteria. For instance, regarding the m parameter in SFRF, in Figure 5 we show that using a lower value as $m = 4$ could provide a little

better edge and detail preservation than $m = 7$.

IV. CONCLUSIONS

In this paper, we propose a simple and effective fuzzy method to reduce Gaussian and impulsive noise from colour images. The method uses one only filtering operation, a weighted averaging, which uses a set of weights computed by a fuzzy rule system. In turn, the fuzzy rule system uses two sources of information on the pixels in each filtering window: (i) their degrees of noisiness (from the impulsive point of view) computed using a statistical method, and (ii) the degrees of similarity between the central pixel and the rest of the pixels in the window. Experimental results show that the method is able to reduce noise and preserve image details, providing competitive results.

REFERENCES

- [1] K. N. Plataniotis, A. N. Venetsanopoulos, *Color Image Processing and Applications*. Springer, Berlin, 2000.
- [2] R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, A.N. Venetsanopoulos, "Vector Filtering for Color Imaging", *IEEE Signal Processing Magazine, Special Issue on Color Image Processing*, vol. 22, no. 1, pp. 74-86, Jan. 2005.
- [3] R. Lukac, K.N. Plataniotis, A taxonomy of color image filtering and enhancement solutions, in *Advances in Imaging and Electron Physics*, (eds.) P.W. Hawkes, Elsevier, **140**, 187-264, 2006.
- [4] Y. Deng, C. Kenney, MS Moore, BS Manjunath, "Peer group filtering and perceptual color image quantization", in *Proceedings of IEEE international symposium on circuits and systems*, vol. 4, 1999, pp. 21-4.



Fig. 4. Outputs for visual comparison: (a) Flower corrupted with $\sigma = 5$ Gaussian and $p = 0.05$ uniform noise, (e) Motorbikes corrupted with $\sigma = 10$ Gaussian and $p = 0.10$ uniform noise, (i) Parrots corrupted with $\sigma = 20$ Gaussian and $p = 0.20$ impulse noise, (m) Lenna corrupted with $\sigma = 30$ Gaussian and $p = 0.30$ impulse noise and corresponding outputs for PGA (b) (f) (j) and (n), FPGA (c) (g) (k) and (o) and SFRF (d) (h) (l) and (p).

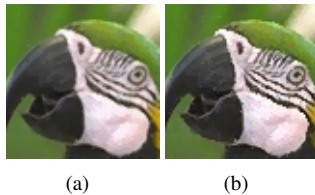


Fig. 5. Results of the SFRF for Parrots image corrupted with $\sigma = 5$ Gaussian and $p = 0.05$ random-value impulse noise: (a) $m = 7$ and (b) $m = 4$.

- [5] G. Hewer, C. Kenney, L. Peterson, A. Van Nevel, "Applied partial differential variational techniques", in Proceedings of International Conference on Image Processing, vol. 3, 1997, pp. 372-375.
- [6] J. Y. F. Ho, "Peer region determination based impulsive noise detection", Proceedings of International Conference on Acoustics, Speech and Signal Processing ICASSP'03, vol. 3, 2003, pp. 713-716.
- [7] C. Kenney, Y. Deng, BS Manjunath, G. Hewer, "Peer group image enhancement", *IEEE Transactions on Image Processing*, vol. 10 no. 2, pp. 326-334, Feb. 2001.
- [8] S. Morillas, V. Gregori, A. Hervás, "Fuzzy peer groups for reducing mixed Gaussian impulse noise from color images", *IEEE Transactions on Image Processing*, vol. 18 no. 7, pp. 1452-1466, Jul. 2008.
- [9] Garnett, R., Huegerich, T., Chui, C., He, W., "A universal noise removal algorithm with an impulse detector", *IEEE Transactions on Image Processing*, vol. 14 no. 11, pp. 1747-1754, Nov. 2005.
- [10] C. Tomasi, R. Manduchi, "Bilateral filter for gray and color images", in Proc. IEEE International Conference Computer Vision, 1998, pp. 839-846.
- [11] M. Elad, "On the origin of bilateral filter and ways to improve it", *IEEE Transactions on Image Processing*, vol. 11 no. 10, pp. 1141-1151, Oct. 2002.
- [12] C.H. Lin, J.S. Tsai, C.T. Chiu, "Switching bilateral filter with texture/noise detector for universal noise removal", *IEEE Transactions on Image Processing*, vol. 19 no. 8, pp. 2307-2320, Sep. 2010.
- [13] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, "Multichannel filters for image processing", *Signal Processing: Image Communication*, vol. 9 no. 2, pp. 143-158, Jan. 1997.
- [14] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Adaptive fuzzy systems for multichannel signal processing, The Proceedings of the IEEE, vol. 87 no. 9, pp. 1601-1622, Sep. 1999.
- [15] Y. Shen, K.E. Barner, "Fuzzy Vector Median Based Surface Smoothing", *IEEE Transactions on Visualization and Computer Graphics*, vol. 10 no. 3, pp. 252-265, May-June 2004.
- [16] Z. Ma, H.R. Wu, D. Feng, "Partition Based Vector Filtering Technique for Suppression of Noise in Digital Color Images", *IEEE Transactions on Image Processing*, vol. 15 no. 8, pp. 2324-2342, Aug. 2006.
- [17] Z. Ma, H.R. Wu, D. Feng, "Fuzzy Vector Partition Filtering Technique for Color Image Restoration", *Computer Vision and Image Understanding*, vol. 107 no. 1-2, pp. 26-37, July-Aug. 2007.
- [18] E. López-Rubio, "Restoration of images corrupted by Gaussian and uniform impulsive noise", *Pattern Recognition*, vol. 43 no. 5, pp. 1835-1846, May 2010.
- [19] D. Keren, A. Gotlib, "Denoising Color Images using regularization and correlation terms", *Journal of Visual Communication and Image Representation*, vol. 9 no. 4, pp. 352-365, Dec. 1998.
- [20] O. Lezoray, A. Elmoataz, S. Boughleux, "Graph regularization for color image processing", *Computer Vision and Image Understanding*, vol. 107 no. 1-2, pp. 38-55, July-Aug. 2007.
- [21] A. Elmoataz, O. Lezoray, S. Boughleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing", *IEEE Transactions on Image Processing*, vol. 17 no. 7, pp. 1047-1060, July 2008.
- [22] P. Blomgren, T. Chan, "Color TV: total variation methods for restoration of vector-valued images", *IEEE Transactions on Image Processing*, vol. 7 no. 3, pp. 304-309, Mar. 1998.
- [23] D. Tschumperlé, R. Deriche, "Vector-valued image regularization with PDEs: A Common framework from different applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27 no. 4, pp. 506-517, April 2005.
- [24] G. Plonka, J. Ma, "Nonlinear regularized reaction-diffusion filters for denoising of images with textures", *IEEE Transactions on Image Processing*, vol. 17 no. 8, pp. 1283-1294, Aug. 2007.
- [25] Y. Xiao, T. Zeng, J. Yu, M.K. Ng, "Restoration of images corrupted by mixed Gaussian-impulse noise via $l_1 - l_0$ minimization", *Pattern Recognition*, vol. 44 no. 8, pp. 1708-1720, Aug. 2011.
- [26] J. Liu, Z. Huan, H. Huan, "Image restoration under mixed noise using globally convex segmentation", *Journal of Visual Communication and Image Representation*, vol. 22 no. 3, pp. 263-270, Apr. 2011.
- [27] X. Zeng, L. Yang, "Mixed impulse and Gaussian noise removal using detail-preserving regularization", *Optical Engineering* vol. 49 no. 9, pp. 097002-1-097002-9, Sep. 2010.
- [28] R. Liu, S. Fu, C. Zhang, "Adaptive mixed image denoising based on image decomposition", *Optical Engineering Letters*, vol. 50 no. 2, pp. 020502-1-020502-3, Feb. 2011.
- [29] O. Ghita, P.F. Whelan, "A new GVF-based image enhancement formulation for use in the presence of mixed noise", *Pattern Recognition*, vol. 43 no. 8, pp. 2646-2658, Aug. 2010.
- [30] B.P. Lamichhane, "Finite element techniques for removing the mixture of Gaussian and impulsive noise", *IEEE Transactions on Image Processing*, vol. 57 no. 7, pp. 2538-2547, Jul. 2009.
- [31] F. Luisier, T. Blu, M. Unser, "Image denoising in mixed Poisson-Gaussian noise", *IEEE Transactions on Image Processing*, vol. 20 no. 3, pp. 696-708, Mar. 2011.
- [32] J.G. Camarena, V. Gregori, S. Morillas, A. Sapena, "Two-step fuzzy logic-based method for impulse noise detection in colour images", *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1842-1849, Oct. 2010.
- [33] V. Gregori, S. Morillas, A. Sapena, "Examples of fuzzy metrics and applications", *Fuzzy Sets and Systems*, vol. 170, no. 1, pp. 95-111, May 2011.
- [34] S. Morillas, V. Gregori, "Robustifying vector median filter", *Sensors*, vol. 11, no. 8, pp. 8115-8126, Aug. 2011.
- [35] K. Passino, S. Yurkovich, *Fuzzy Control*, Addison Wesley, Menlo Park, California, 1998.
- [36] D. Driankov, H. Hellendoorn, M. Reinfrank, *An Introduction to Fuzzy Control*, 2nd Edition, Springer, 1996.
- [37] E. Cox, *The Fuzzy Systems Handbook*, 2nd Ed., Academic Press, 1999.
- [38] M. Szczepanski, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, "On the distance function approach to color image enhancement", *Discrete Applied Mathematics* no. 139, pp. 283-305, 2004.
- [39] KODAK test images database, <http://r0k.us/graphics/kodak/>.