Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 158 (2007) 270–283
FUZZY sets and systems
www.elsevier.com/locate/fss

Fuzzy random impulse noise reduction method

Stefan Schulte*, Valérie De Witte, Mike Nachtegael, Dietrich Van der Weken, Etienne E. Kerre

Ghent University, Department of Applied Mathematics and Computer Science, Fuzziness and Uncertainty Modelling Research Unit, Krijgslaan 281 (Building S9), B-9000 Gent, Belgium

Available online 19 October 2006

Abstract

A new two-step fuzzy filter that adopts a fuzzy logic approach for the enhancement of images corrupted with impulse noise is presented in this paper. The filtering method (entitled as Fuzzy Random Impulse Noise Reduction method (FRINR)) consists of a fuzzy detection mechanism and a fuzzy filtering method to remove (random-valued) impulse noise from corrupted images. Based on the criteria of peak-signal-to-noise-ratio (PSNR) and subjective evaluations we have found experimentally, that the proposed method provides a significant improvement on other state-of-the-art methods.

Keywords: Fuzzy filter; Impulse noise; Image restoration; α-Stable noise

1. Introduction

Fuzzy techniques have already been applied in several domains of image processing e.g. filtering, interpolation [29] and morphology [18,19]. They also have numerous practical applications (e.g., in industrial and medical image processing [20,3]). A major problem of many image processing techniques is that they cannot work well in a noisy environment, so that a preprocessing module became necessary. In this paper, we will focus on fuzzy techniques for digital image filtering. More specifically we will concentrate on a fuzzy logic approach for the enhancement of images corrupted with impulse noise. Images are often corrupted with impulse noise due to a noisy sensor or channel transmission errors. The main goal of impulse noise reduction methods is to suppress the noise while preserving the fine texture and edge elements. Nonlinear techniques have been found to provide more satisfactory results in comparison to linear methods.

A number of nonlinear approaches have been already developed for impulse noise removal, for example the well-known fuzzy inference rule by else-action filters (FIRE). These filters try to calculate positive and negative correction terms in order to express the degree of noise for a certain pixel. We distinguish three FIRE filters: the normal FIRE from [23], the dual step FIRE from [24] (DS-FIRE) and the piecewise linear FIRE from [21] (PWLFIRE). The adaptive weighted fuzzy mean filter [17,16] (AWFM), the histogram adaptive fuzzy filter [30,31] (HAF), the intelligent image agent based on soft-computing techniques [8] (IIA) and the adaptive fuzzy switching filter (AFSF), which uses the

* Corresponding author. Tel.: +32 9 264 47 65; fax: +32 9 264 49 95.
E-mail addresses: Stefan.Schulte@UGent.be (S. Schulte), Valerie.Dewitte@UGent.be (V. De Witte), Mike.Nachtegael@UGent.be (M. Nachtegael), Dietrich.VanderWeken@UGent.be (D. Van der Weken), Etienne.Kerre@UGent.be (E.E. Kerre).
URL: http://www.fuzzy.Ugent.be (S. Schulte).

0165-0114/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2006.10.010

maximum–minimum exclusive median filter [32], are other examples of the state-of-the-art methods. These fuzzy filters are mainly developed for images corrupted with fat-tailed noise like impulse noise. They are also able to outperform rank-order filter schemes (such as the median filter). Examples of real impulse noise include atmospheric noise, cellular communication, underwater acoustics and moving traffic. Recently, it has been shown that α-stable ($0 < \alpha \leqslant 2$) distributions can approximate impulse noise more accurately than other models [13,2]. The goal of this paper is to develop an impulse noise reduction method, which performs well for this, more realistic, kind of randomly valued impulse noise model.

The structure of the paper is organized as follows: in Section 2 we explain the detection method. The filtering method of the new filter (FRINR) is discussed in Section 3. In Section 4 we present several experimental results. These results are discussed in detail, and are compared to those obtained by other filters. Some final conclusions are drawn in Section 5.

1.1. Impulse noise models

In many current impulse noise models for images, corrupted pixels are often replaced with values equal to or near the maximum or minimum intensity values of the allowable dynamic range. This typically corresponds to fixed values near 0 or 255 (for a 8-bit image). In this paper, we consider a more general noise model in which a noisy pixel is taken as an arbitrary value in the dynamic range according to some underlying probability distribution. Let $O(i, j)$ and $A(i, j)$ denote the intensity value, at position (i, j), of the original and the noisy image, respectively. Then, for an impulse noise model with error probability pr, we have

$$A(i, j) = \begin{cases} O(i, j) & \text{with probability } 1 - \text{pr}, \\ \eta(i, j) & \text{with probability pr}, \end{cases} \tag{1}$$

where $\eta(i, j)$ is an identically distributed, independent random process with an arbitrary underlying probability density function.

Recently, it has been shown that an α-stable distribution can approximate impulse noise more accurately than other models. Therefore we also consider this type of randomly distributed impulse noise. The parameter α controls the degree of impulsiveness and the impulsiveness increases as α decreases. The Gaussian ($\alpha = 2$) and the Cauchy ($\alpha = 1$) distributions are the only symmetric α-stable distributions that have closed-form probability density functions. A symmetric α-stable ($S\alpha S$) random variable is only described by its characteristic function:

$$\varphi(t) = \exp(j\theta t - \gamma|t|^{\alpha}),$$

where $j \in \mathbb{C}$ is the imaginary unit, $\theta \in \mathbb{R}$ is the location parameter (centrality), $\gamma \in \mathbb{R}$ is the dispersion of the distribution and $\alpha \in [0, 2[$, which controls the heaviness of the tails, is the characteristic exponent. For more background information about the α-stable noise model we refer to [13,2].

1.2. The FRINR method

The Fuzzy Random Impulse Noise Reduction method (FRINR) consists of two separated phases: the detection phase and the filtering phase. The detection phase is a combination of two fuzzy algorithms, which are complementary to each other and which are combined together to receive a more robust detection method. Both algorithms use fuzzy rules [27] to determine whether a pixel is corrupted with impulse noise or not. After the application of the two fuzzy algorithms, our fuzzy filtering technique focuses only on those pixels that are detected by both algorithms, i.e. the filtering is concentrated on the real impulse noise pixels. The filtering method consists of a fuzzy averaging where the weights are constructed using a predefined fuzzy set.

2. Detection method

The proposed detection method is composed of two subunits that are both used to define corrupted impulse noise pixels. The first subunit investigates the neighbourhood around a pixel to conclude if the pixel can be considered as impulse noise or not. The second subunit uses fuzzy gradient values [28] to determine the degree in which a pixel can be considered as impulse noise and the degree in which a pixel can be considered as noise free.

2.1. First detection unit

We consider a two dimensional input image denoted as A. The goal of the first detection unit is to determine if a certain image pixel $A(i, j)$, at position (i, j), is corrupted with impulse noise or not. Therefore we observe the elements in a $(2K + 1) \times (2K + 1)$ (with $K \geqslant 1$) window centred around $A(i, j)$. Next, we calculate the mean differences in this window denoted as $g(i, j)$:

$$g(i, j) = \frac{\sum_{k=-K}^{K} \sum_{l=-K}^{K} |A(i + k, j + l) - A(i, j)|}{(2K + 1)^2 - 1}. \tag{2}$$

Corrupted impulse noise pixels generally cause large $g(i, j)$ values, because impulse noise pixels normally occur as outliners in a small neighbourhood (during this paper we consider $K = 1$) around the pixel. On the other hand we also found that the $g(i, j)$-value could be relatively large in case of an edge pixel. Therefore we have considered the following two values denoted as $obs_1(i, j)$ and $obs_2(i, j)$:

$$obs_1(i, j) = \frac{\sum_{k=-K}^{K} \sum_{l=-K}^{K} g(i + k, j + l)}{(2K + 1)^2}, \tag{3}$$

$$obs_2(i, j) = g(i, j). \tag{4}$$

If both values ($obs_1(i, j)$ and $obs_2(i, j)$) are large, then the pixel can be considered as an edge pixel instead of a noisy one. So when the two values ($obs_1(i, j)$ and $obs_2(i, j)$) are very similar we conclude that the pixel is noise free. Otherwise, if the difference between $obs_1(i, j)$ and $obs_2(i, j)$ is large then we consider the pixel as noisy. This can be implemented by the following fuzzy rule:

Fuzzy Rule 1. *Defining when a central pixel $A(i, j)$ is corrupted with impulse noise*:
IF $|obs_1(i, j) - obs_2(i, j)|$ *is large*
THEN *the central pixel $A(i, j)$ is an impulse noise pixel*

In this rule, *large* can be represented as a fuzzy set [11]. A fuzzy set in turn can be represented by a membership function. An example of a membership function LARGE (for the corresponding fuzzy set *large*), which is entitled as μ_{large}, is pictured in Fig. 1. From such functions we can derive membership degrees. If the difference $|obs_1(i, j) - obs_2(i, j)|$ for example has a membership degree one (or zero) in the fuzzy set *large*, it means that this difference is considered as large (or not large) for sure. Membership degrees between zero and one indicate that we do not know for sure if such difference is large or not, so that the difference is large to a certain degree. For more background information about fuzzy set theory we refer to Kerre [11]. In Fig. 1 we see that we have to determine two important parameters a and b. The parameter a is equal to the lowest $g(i + k, j + l)$ value in the $(2K + 1) \times (2K + 1)$ window around the central pixel, i.e.

$$a(i, j) = \min_{k,l \in \{-K,\ldots,+K\}} (g(i + k, j + l)). \tag{5}$$

So, $a(i, j)$ corresponds to the $g(i + k, j + l)$ coming from the most homogeneous region around $A(i, j)$, which should correspond to the region with the smallest amount of impulse noise pixels. Experimental results have shown that the best choice for parameter $b(i, j)$ is $b(i, j) = a(i, j) + 0.2\,a(i, j)$, i.e. the larger the parameter a, the larger the uncertainty interval $[a, b]$ should be.

So, the outputs of the first detection method are the membership degrees in the fuzzy set *impulse noise* for each pixel separately. The membership function that represents this fuzzy set is denoted as μ_{impulse}. The corresponding membership degrees ($\in [0, 1]$) are calculated using Fuzzy Rule 1. The activation degree of this rule is used to determine the membership degree μ_{impulse}, i.e. $\mu_{\text{impulse}}(A(i, j)) = \mu_{\text{large}}(|obs_1(i, j) - obs_2(i, j|))$.

2.2. Second detection unit

As for the first detection unit, also for the second one we want to calculate the degree in which a certain pixel $A(i, j)$ can be considered as impulse noise. Both units are complementary to each other, i.e. by combining them we receive a more robust detection method that improves the global performance.

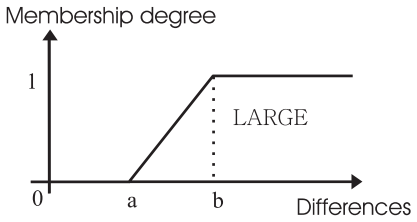

Fig. 1. The membership function LARGE denoted as μ_{large}.

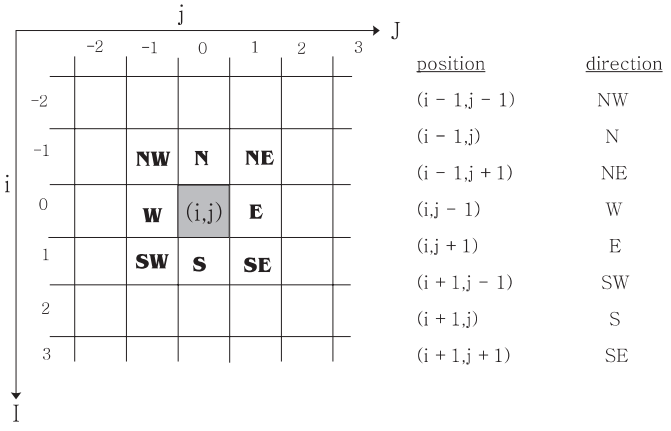

Fig. 2. A neighbourhood of a central pixel $A(i, j)$.

We again use a $(2K + 1) \times (2K + 1)$ (we have used $K = 1$) neighbourhood around $A(i, j)$ as illustrated in Fig. 2. Each of the eight neighbours of $A(i, j)$ corresponds to one direction {North West (NW), North (N), North East (NE), East (E), South East (SE), South (S), South West (SW), West (W)}. Next, we define the gradient value $\bigtriangledown_D A(i, j)$ of pixel position (i, j) in direction D, which corresponds to a certain position (shown in Fig. 2) as

$$\bigtriangledown_{(k,l)} A(i, j) = A(i + k, j + l) - A(i, j) \quad \text{with } k, l \in \{-K, \ldots, K\}, \tag{6}$$

where the pair (k, l) corresponds to one of the eight directions and (i, j) is called the centre of the gradient. The eight gradient values (associated to the eight directions or neighbours) are called the basic gradient values. There are two cases where large gradient values occur: (1) if one of the two pixels is corrupted with impulse noise or (2) when an edge is presented. Here we only want to detect the first case and therefore we use not only one basic gradient value for each direction but also two related gradient values (defined in the same direction). These two related gradient values in the same direction as the basic gradient are determined by the centres making a right-angle with the direction of the corresponding basic gradient. This is illustrated in Fig. 3 for the NW-direction (i.e. for $(k, l) = (-1, -1)$). The basic gradient and the two related gradient values at position (i, j) are defined as $\bigtriangledown_{(-1,-1)} A(i, j)$, $\bigtriangledown_{(-1,-1)} A(i - 1, j + 1)$ and $\bigtriangledown_{(-1,-1)} A(i + 1, j - 1)$, respectively.

In Table 1 we give an overview of the involved gradient values: each direction D (column 1) corresponds to a position (Fig. 2) w.r.t. a central position. Column 2 states the basic gradient for each direction, column 3 lists the two related gradients.

For each direction we will finally calculate a membership degree in the fuzzy set *impulse noise* (denoted as $\gamma^D_{\text{impulse}}$ for direction D) and the membership degree in the fuzzy set *impulse noise free* (denoted as γ^D_{free} for direction D). This is realized by the following Fuzzy Rules 2 and 3.

Fuzzy Rule 2. *Defining when a central pixel $A(i, j)$ is corrupted with impulse noise for a certain direction D:*
IF ($|\bigtriangledown_D A(i, j)|$ *is* **not** *large*) AND ($|\bigtriangledown'_D A(i, j)|$ *is large*) AND ($|\bigtriangledown''_D A(i, j)|$ *is large*)

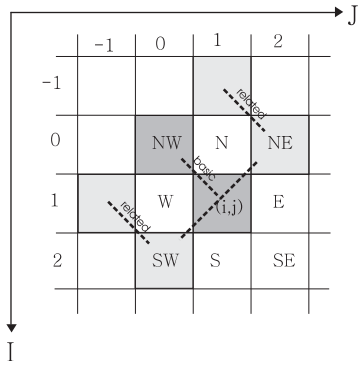

Fig. 3. Involved centres for the calculation of the related gradient values in the *NW*-direction.

Table 1
Involved gradient values to calculate the fuzzy gradient

D	Basic gradient	Related gradients
NW	$\nabla_{\mathrm{NW}}A(i,j)$	$\nabla_{\mathrm{NW}}A(i+1,j-1), \nabla_{\mathrm{NW}}A(i-1,j+1)$
N	$\nabla_{\mathrm{N}}A(i,j)$	$\nabla_{\mathrm{N}}A(i,j-1), \nabla_{\mathrm{N}}A(i,j+1)$
NE	$\nabla_{\mathrm{NE}}A(i,j)$	$\nabla_{\mathrm{NE}}A(i-1,j-1), \nabla_{\mathrm{NE}}A(i+1,j+1)$
E	$\nabla_{\mathrm{E}}A(i,j)$	$\nabla_{\mathrm{E}}A(i-1,j), \nabla_{\mathrm{E}}A(i+1,j)$
SE	$\nabla_{\mathrm{SE}}A(i,j)$	$\nabla_{\mathrm{SE}}A(i-1,j+1), \nabla_{\mathrm{SE}}A(i+1,j-1)$
S	$\nabla_{\mathrm{S}}A(i,j)$	$\nabla_{\mathrm{S}}A(i,j-1), \nabla_{\mathrm{S}}A(i,j+1)$
SW	$\nabla_{\mathrm{SW}}A(i,j)$	$\nabla_{\mathrm{SW}}A(i-1,j-1), \nabla_{\mathrm{SW}}A(i+1,j+1)$
W	$\nabla_{\mathrm{W}}A(i,j)$	$\nabla_{\mathrm{W}}A(i-1,j), \nabla_{\mathrm{W}}A(i+1,j)$

OR
($|\nabla_D A(i,j)|$ *is large*) AND ($|\nabla'_D A(i,j)|$ *is* **not** *large*) AND ($|\nabla''_D A(i,j)|$ *is* **not** *large*)
OR
($|\nabla_D A(i,j)|$ *is large*) AND ($|\nabla'_D A(i,j)|$ *is large*) AND ($|\nabla''_D A(i,j)|$ *is* **not** *large*)
OR
($|\nabla_D A(i,j)|$ *is large*) AND ($|\nabla'_D A(i,j)|$ *is* **not** *large*) AND ($|\nabla''_D A(i,j)|$ *is large*)
THEN *the central pixel* $A(i,j)$ *is an impulse noise pixel in direction* D

Fuzzy Rule 3. *Defining when a central pixel* $A(i,j)$ *is not corrupted with impulse noise for a certain direction* D:
IF ($|\nabla_D A(i,j)|$ *is large*) AND ($|\nabla'_D A(i,j)|$ *is large*) AND ($|\nabla''_D A(i,j)|$ *is large*)
OR
($|\nabla_D A(i,j)|$ *is not large*) AND ($|\nabla'_D A(i,j)|$ *is not large*) AND ($|\nabla''_D A(i,j)|$ *is not large*)
THEN *the central pixel* $A(i,j)$ *is impulse noise free in direction* D

We have denoted the basic gradient as $\nabla_D A(i,j)$, while the two related gradient values were entitled as $\nabla'_D A(i,j)$ and $\nabla''_D A(i,j)$, respectively. Fuzzy Rule 2 determines when a certain pixel can be observed as impulse noise or not. This rule contains conjunctions and disjunctions. In fuzzy logic triangular norms and co-norms are used to represent conjunction (roughly the equivalent of the AND operator) and disjunction (roughly the equivalent of the OR operator) [5]. Two well-known triangular norms (together with their dual co-norms) are the product (probabilistic sum) and the minimum (maximum). In fuzzy logic involutive negators are commonly used to represent negations. We use the standard negator $N_s(x) = 1 - x$, with $x \in [0,1]$. We illustrate the concept of triangular norm and standard negator with a short example: the fuzzification of the following statement, i.e. we assign a value in the unit interval to indicate

the truthness of this statement (1 indicates that this statement is true for sure and 0 indicates that this statement is false for sure), "($|\nabla_D\ A(i,j)|$ *is not large*) *AND* ($|\nabla'\ A(i,j)|$ *is large*) *AND* ($|\nabla''\ A(i,j)|$ *is large)*" is calculated by $(1-\mu_{\text{large}}(|\nabla_D\ A(i,j)|))\cdot\mu_{\text{large}}(|\nabla'_D\ A(i,j)|)\cdot\mu_{\text{large}}(|\nabla''_D\ A(i,j)|)$, where we used the "product" triangular norm and where μ_{large} has the same graph as in Fig. 1 using the following parameters a and b:

$$a(i,j)=\frac{(\sum_{k=-K}^{K}\sum_{l=-K}^{K}g(i+k,j+l))-g(i,j)}{(2K+1)^2-1},$$

$$b(i,j)=a(i,j)+a(i,j)*0.2. \tag{7}$$

So, $a(i,j)$ corresponds to the average of the $g(i+k,j+l)$ values. By using these parameters we have managed the incorporation of regions containing edges, because these non-homogeneous regions will cause higher parameters so that our detection method is adapted to such situations. Gradient values in non-homogeneous regions are labelled as large if they are large in comparison to their neighbours. In homogeneous regions we will get smaller values of $a(i,j)$ and $b(i,j)$, which will cause a much stronger detection method.

The outputs of the second detection unit are the eight membership degrees in the fuzzy set *impulse noise* for the eight directions around a certain position (i,j), i.e. the degrees $\gamma^D_{\text{impulse}}A(i,j)$ and the eight membership degrees in the fuzzy set *impulse noise free* for the eight directions around a certain position (i,j) , i.e. the degrees $\gamma^D_{\text{free}}A(i,j)$. The degrees $\gamma^D_{\text{impulse}}A(i,j)$ and $\gamma^D_{\text{free}}A(i,j)$ are calculated using Fuzzy Rule 2 and 3, respectively.

3. Filtering method

We combine both detection units to determine the pixels where the filtering method will be applied, i.e. we will apply the filtering method on pixels that are determined as noisy in both units. Pixels having a non-zero membership degree in the fuzzy set *impulse noise* for the first detection unit are considered as noisy, i.e. $\mu_{\text{impulse}}(A(i,j))>0$. In the second detection unit we consider two fuzzy sets namely *impulse noise* and *impulse noise free* in order to decide if a pixel is considered as noisy or not. Here we have decided that if $\sum_{D\in\{N,\ldots,S\}}\gamma^D_{\text{impulse}}A(i,j)\geqslant\sum_{D\in\{N,\ldots,S\}}\gamma^D_{\text{free}}A(i,j)$ then impulse noise is considered at $A(i,j)$. So, the filtering method will be applied to pixels where both restrictions are satisfied, i.e.

$$\mu_{\text{impulse}}(A(i,j))>0,$$

$$\sum_{D\in\{N,\ldots,S\}}\gamma^D_{\text{impulse}}A(i,j)\geqslant\sum_{D\in\{N,\ldots,S\}}\gamma^D_{\text{free}}A(i,j). \tag{8}$$

The output of the filtering method for the input pixel $A(i,j)$ is denoted as $F(i,j)$ and is calculated as follows:

$$F(i,j)=(1-\lambda(i,j))\frac{\sum_{k=-L}^{L}\sum_{l=-L}^{L}A(i+k,j+l)w(i+k,j+l)}{\sum_{k=-L}^{L}\sum_{l=-L}^{L}w(i+l,j+k)}+\lambda(i,j)A(i,j). \tag{9}$$

The filtering method uses a $(2L+1)\times(2L+1)$ (not necessary equal to K) neighbourhood around $A(i,j)$ as shown in expression 9. Each $A(i+k,j+l)$ is multiplied by a corresponding weight $w(i+k,j+l)$ indicating in which degree the pixel should be used (explained later) to filter the central pixel. The parameter $\lambda(i,j)$ is finally used to control the amount of correction.

To determine the weights, we have used the pixels from the $(2K+1)\times(2K+1)$ neighbourhood around $A(i,j)$ (with the same K as in the detection method). Those pixels are then sorted so that the window is denoted as $[x_1,x_2,\ldots,x_{(2K+1)^2}]$ with x_1 and $x_{(2K+1)^2}$ the lowest and highest intensity value from the corresponding window, respectively.

We do not replace the corrupted pixels by their median, but nevertheless we try to use the other pixels in the neighbourhood as well. When the $(2K+1)\times(2K+1)$ neighbourhood is very homogeneous we know that median based algorithms are working very well, but when we have non-homogeneous neighbourhoods it is better to incorporate the knowledge of the other pixels as well to improve the filtering performance. Therefore we have constructed a fuzzy set called *similar* to calculate the corresponding weights $w(i+k,j+l)$. This fuzzy set is represented by the membership function SIMILAR, pictured in Fig. 4 and denoted as μ_{sim}. The membership value indicates in which degree a certain

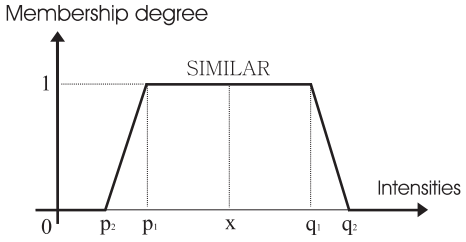

Fig. 4. The membership function SIMILAR denoted as μ_{sim}.

intensity value can be observed as similar to the observed neighbourhood. Pixels having a degree of one (zero) are (not) similar for sure to the corresponding neighbourhood. So the weight $w(i+k, j+l)$ is defined as $w(i+k, j+l) = \mu_{\text{sim}}(A(i+k, j+l))$.

The membership function is determined by means of four parameters. To define these parameters (p_1, p_2, q_1 and q_2) we first have calculated the mean differences ρ, i.e.

$$\rho(i, j) = \frac{\sum_{k=2}^{(2K+1)^2}(x_k - x_{k-1})}{(2K+1)^2 - 1}. \tag{10}$$

Using this $\rho(i, j)$ we have determined the parameters as follows:

$$x(i, j) = \underbrace{\textit{median}}_{k,l\in\{-K,\ldots,+K\}} (A(i+k, j+l)),$$

$$p_1(i, j) = x(i, j) - \rho(i, j), \quad p_2(i, j) = x(i, j) - 1.1\rho(i, j),$$

$$q_1(i, j) = x(i, j) + \rho(i, j), \quad q_2(i, j) = x(i, j) + 1.1\rho(i, j). \tag{11}$$

Homogeneous regions will have very similar pixels, which will cause very low $\rho(i, j)$ values, because the differences between the pixel intensity values will be low. In these situations we will receive parameters ($p_1(i, j)$, $p_2(i, j)$, $q_1(i, j)$ and $q_2(i, j)$) that are very close to the median (the centre of the fuzzy set) and therefore the filtering performance will be similar to median based filters. On the other hand we will get an improvement in cases of non-homogeneous windows, because we still incorporate the information of all the neighbouring pixels around $A(i, j)$ as well.

Finally, we will reduce the correction process for the central pixels $A(i, j)$ having a very high weight $w(i, j)$ because if the corresponding weight is very high, it means that these pixels seem to be similar to their neighbourhood unless the detection. Therefore we define the parameter $\lambda(i, j)$ to be equal to the weight of the centre, i.e. $\lambda(i, j) = w(i, j)$.

4. Results

In this section, we compare the performance of our method (FRINR) with some state-of-the-art methods for reducing impulse noise from digital grayscale images. The experiments were carried out on the well-known test images: "Lena" and "Boats" that are, respectively, 512×512 pixel sized and 576×720 pixel sized. The objective quantitative measures used for comparison are the mean square error (denoted as MSE) and the peak signal-to-noise ratio (PSNR) between the original and restored images, defined by

$$MSE(F, O) = \frac{1}{N\,M} \sum_{i=1}^{N} \sum_{j=1}^{M} [\mathrm{O}(i, j) - F(i, j)]^2, \tag{12}$$

$$PSNR(F, O) = 10 \log_{10} \frac{S^2}{MSE(F, O)}, \tag{13}$$

Table 2
Comparative results in PSNR of different filtering methods for various percentages of randomly valued impulse noise for the (512 × 512) Lena image

Method	Noise density								
	0.03	0.05	0.10	0.15	0.20	0.25	0.30	0.40	0.50
Noisy	24.6	22.3	19.2	17.3	16.1	15.3	14.5	13.3	12.3
CWM	37.8	37.2	34.3	32.3	31.4	30.8	29.7	28.2	26.1
TSM	35.1	34.6	33.8	32.4	31.7	30.9	29.8	28.3	26.3
LUM	38.8	37.9	33.8	31.5	30.8	29.8	29.2	27.8	25.8
MFF	31.1	30.6	29.8	28.8	28.0	26.9	25.9	23.6	21.3
ATMED	33.9	33.3	31.8	30.4	28.8	27.3	25.6	22.8	20.4
GMED	35.0	34.6	33.6	32.8	31.5	30.0	28.1	24.7	21.7
TMAV	34.8	34.3	32.9	31.5	29.7	28.0	26.3	23.4	20.8
AFSF	38.9	37.4	34.9	33.2	31.0	29.3	26.9	23.7	20.9
IIA	34.7	32.9	30.1	28.3	26.5	25.0	23.6	21.3	19.1
FSB	34.2	33.9	33.1	32.4	31.3	29.8	28.7	24.7	21.7
IFCF	34.2	33.7	32.7	31.9	30.9	30.0	28.7	26.2	23.5
MIFCF	34.1	33.6	32.3	31.4	30.3	29.2	27.8	25.1	22.3
EIFCF	34.1	33.7	32.7	31.9	30.9	29.9	28.6	26.0	23.4
SSFCF	33.1	32.7	31.8	30.8	29.5	28.1	26.4	23.4	20.6
FIRE	36.8	34.6	31.2	29.8	27.3	25.5	24.0	21.4	19.0
PWLFIRE	30.9	28.5	25.3	23.7	21.9	20.8	19.6	17.8	16.2
DSFIRE	31.2	29.2	26.2	24.5	23.0	22.1	21.1	19.5	18.0
FMF	39.8	38.1	35.2	33.8	31.6	29.9	27.8	25.3	22.7
HAF	33.4	32.7	30.2	28.8	26.4	24.8	23.4	21.1	19.3
AWFM	31.5	30.8	30.0	28.0	25.9	24.3	22.8	20.5	18.5
FIDRM	34.3	32.9	31.1	29.3	28.0	26.2	23.9	21.1	18.9
FRINR	42.4	41.1	38.5	36.3	35.0	33.9	32.8	30.7	27.0

where O is the original image, F the restored image of size NM and S the maximum possible intensity value (with 8-bit the maximum will be 255).

We have compared our new fuzzy filter entitled Fuzzy Random Impulse Noise Reduction method (FRINR) with the following state-of-the-art filter techniques:

- The fuzzy inference rule by else-action filters (FIRE). The idea behind these filters is that they try to calculate positive and negative correction terms in order to express the degree of noise for a certain pixel. We distinguish four FIRE filters: the normal FIRE from [23], the dual step FIRE (DSFIRE) from [24], the piecewise linear FIRE (PWLFIRE) from [21] and the multipass fuzzy filter (MFF) that contains three sub-filters based on the previous FIRE filters [22].
- The fuzzy median filter (FMF) from [1].
- The adaptive weighted fuzzy mean filter (AWFM) from [16,17].
- The fuzzy control based filters [6,7]. These filters correct a certain central pixel value according to some features of some luminance (pixel values) differences between the central pixel value and some neighbouring pixel values. Here we use the iterative fuzzy control based filter (IFCF), the modified IFCF (MIFCF), the extended IFCF (EIFCF) and the sharpening smoothing fuzzy control based filter (SSFCF).
- The asymmetrical triangular fuzzy filter with median centre (ATMED), the Gaussian fuzzy filter with median centre from (GMED) and the symmetrical triangle fuzzy filter with average centre from [14,15].
- The histogram adaptive fuzzy filter [30,31], which of course uses the histogram of an image (HAF).
- The intelligent image agent based on soft-computing technique (IIA) from [8].
- The adaptive fuzzy switching filter (AFSF) from [32], which uses the maximum–minimum exclusive median filter.
- The fuzzy similarity filter (FSB) from [10], where the local similarity between some patterns is used.
- The classical median based filters. The centre weighted median (CWM) from [12], the tri-state median filter (TSM) from [4] and the lower–upper–middle filter (LUM) from [9].
- The fuzzy impulse noise detection and reducing method (FIDRM) from [25,26].

Table 3
PSNR results for the (512 × 512-) Lena image corrupted with an α-stable impulse noise

Method	α					
	0.8	0.7	0.6	0.5	0.4	0.3
Noisy	22.2	20.3	18.3	16.4	14.4	12.1
CWM	34.9	34.0	32.1	31.4	30.9	30.3
TSM	33.8	33.6	31.5	31.0	30.3	29.0
LUM	34.3	33.9	31.0	30.8	30.1	29.7
MFF	30.3	30.0	29.3	28.6	27.4	25.2
ATMED	33.2	32.6	31.8	31.0	29.9	28.6
GMED	34.1	33.7	33.2	32.4	31.1	28.6
TMAV	34.0	33.5	32.7	31.5	29.6	26.7
AFSF	35.6	34.3	33.3	32.1	31.1	30.0
IIA	32.0	30.7	29.2	28.5	28.1	27.2
FSB	33.7	33.2	32.6	31.9	30.7	28.2
IFCF	33.4	32.9	32.4	31.7	30.5	28.2
MIFCF	33.3	32.8	32.0	31.0	29.3	26.6
EIFCF	33.5	33.0	32.4	31.6	30.4	28.1
SSFCF	32.7	31.8	31.0	29.9	28.1	24.8
FIRE	30.7	29.6	28.5	27.3	25.6	22.9
PWLFIRE	28.6	27.6	26.4	25.4	23.7	21.2
DSFIRE	29.9	28.6	27.5	26.7	25.7	25.1
FMF	35.7	34.5	33.5	32.2	30.8	29.1
HAF	33.0	32.4	31.6	30.6	29.3	28.2
AWFM	30.7	30.5	30.2	29.9	28.6	28.0
FIDRM	32.9	31.9	30.7	29.0	28.1	26.6
FRINR	37.5	36.8	36.2	35.3	33.8	32.4

Table 4
Comparative results in PSNR of different filtering methods for various percentages of randomly valued impulse noise for the (576 × 720) Boats image

Method	Noise density					α			
	0.03	0.15	0.30	0.40	0.50	0.30	0.40	0.50	0.60
Noisy	24.5	17.5	14.5	13.2	12.3	12.5	14.4	16.3	18.2
CWM	36.3	31.8	28.8	27.2	24.3	29.4	30.4	32.0	33.1
TSM	37.9	32.2	28.3	24.5	23.5	27.2	29.8	31.7	32.9
LUM	34.2	31.7	27.9	27.3	24.4	27.9	30.6	32.2	32.8
MFF	30.7	28.6	25.5	23.2	20.8	25.3	27.6	28.2	29.3
ATMED	33.0	29.6	25.0	22.2	19.9	27.9	29.2	30.3	31.1
GMED	33.8	31.7	27.3	24.1	21.2	27.9	30.4	31.6	32.3
TMAV	33.6	30.6	25.8	22.9	20.5	26.3	29.1	30.8	31.8
AFSF	37.4	31.9	26.1	22.9	20.4	29.0	30.3	31.7	32.6
IIA	32.7	25.7	21.4	19.5	17.9	26.3	27.3	27.8	28.7
FSB	33.0	31.3	27.3	24.1	21.1	27.6	30.0	31.1	31.7
IFCF	33.1	31.0	28.0	25.6	23.0	27.5	29.7	30.8	31.6
MIFCF	33.4	30.7	27.3	24.6	21.8	25.9	28.7	30.4	31.4
EIFCF	32.9	31.0	28.0	25.5	22.8	27.3	29.6	30.7	31.5
SSFCF	32.0	30.0	25.8	22.9	20.1	24.3	27.5	29.3	30.2
FIRE	36.2	28.8	23.7	21.1	18.8	22.8	25.7	27.4	28.6
PWLFIRE	30.9	23.5	19.6	17.7	16.1	21.2	23.6	25.2	26.4
DSFIRE	31.4	24.7	21.2	19.4	17.8	25.0	25.8	26.6	27.5
FMF	38.4	32.4	26.8	23.3	21.8	27.5	30.2	31.8	32.7
HAF	32.4	27.8	23.1	20.8	18.9	27.4	28.6	29.7	30.6
AWFM	30.3	27.8	23.8	21.2	19.0	27.8	28.4	28.9	29.3
FIDRM	32.4	29.3	26.7	25.5	23.0	26.7	27.4	28.8	30.1
FRINR	42.7	36.0	31.8	29.6	26.7	30.9	32.7	33.8	34.7

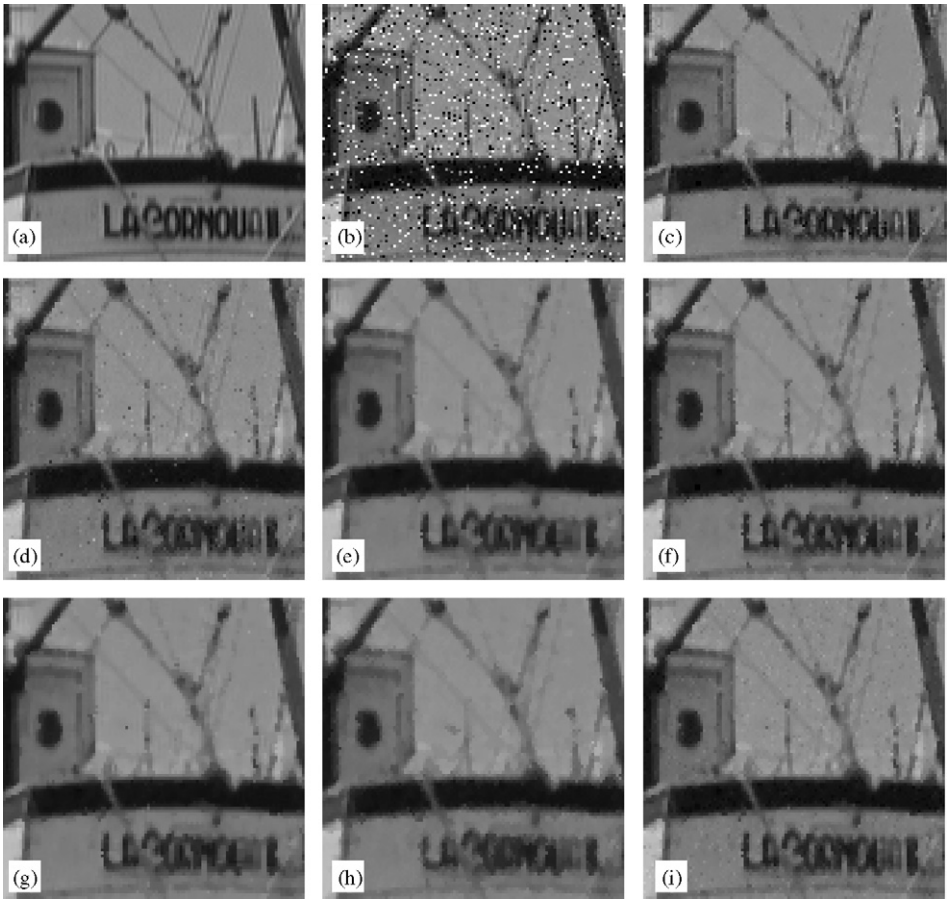

Fig. 5. (a) A part of the original Boats image. (b) The part corrupted with an α-stable noise with $\alpha = 0.5$ (PSNR: 16.3 dB). (c) After the proposed FRINR method (PSNR: 33.8 dB). (d) After the AFSF (PSNR: 31.7 dB). (e) After the GMED (PSNR: 31.6 dB). (f) After the CWM (PSNR: 32.0 dB). (g) After the LUM (PSNR: 32.2 dB). (h) After the IFCF (PSNR: 30.8 dB). (i) After the FMF (PSNR: 31.8 dB).

4.1. Numerical results

Both test images ("Lena" and "Boats") were corrupted with random valued impulse with noise levels varying from 3% (low noise ratio) to 50% (extreme high noise ratio) and with the more realistic α-stable noise where α varied from 0.3 to 0.8. The numerical results for the "Lena" image are shown in Tables 2 and 3 for the random valued impulse noise and α-stable impulse noise, respectively. In Table 4 we have pictured both kinds of noise for the "Boats" image. It is obvious that the proposed method consistently works well for different test images degraded at different noise ratios, providing substantial improvement over the other filters.

We also have to mention that each method is applied several times until we receive the best PSNR value. The proposed method FRINR is generally applied 4 times with corresponding window sizes $K = 1$ and $L = 2$. For the CWM, TSM and LUM methods we always have used the optimal parameters and the window size (3×3, 5×5 or 7×7) that received the best PSNR result.

We see that the PSNR value for the proposed method is better than the second best filters (FMF, CWM, TSM or LUM) by a factor of 0.94 for low as well as for high random valued impulse noise.

4.2. Visual results

Fig. 5 shows the restoration results of different filters for the "Boats" image. Fig. 5(a) and (b) show a magnified part of the original image and the image part corrupted with an α-stable impulse noise, with $\alpha = 0.5$. It can be seen that

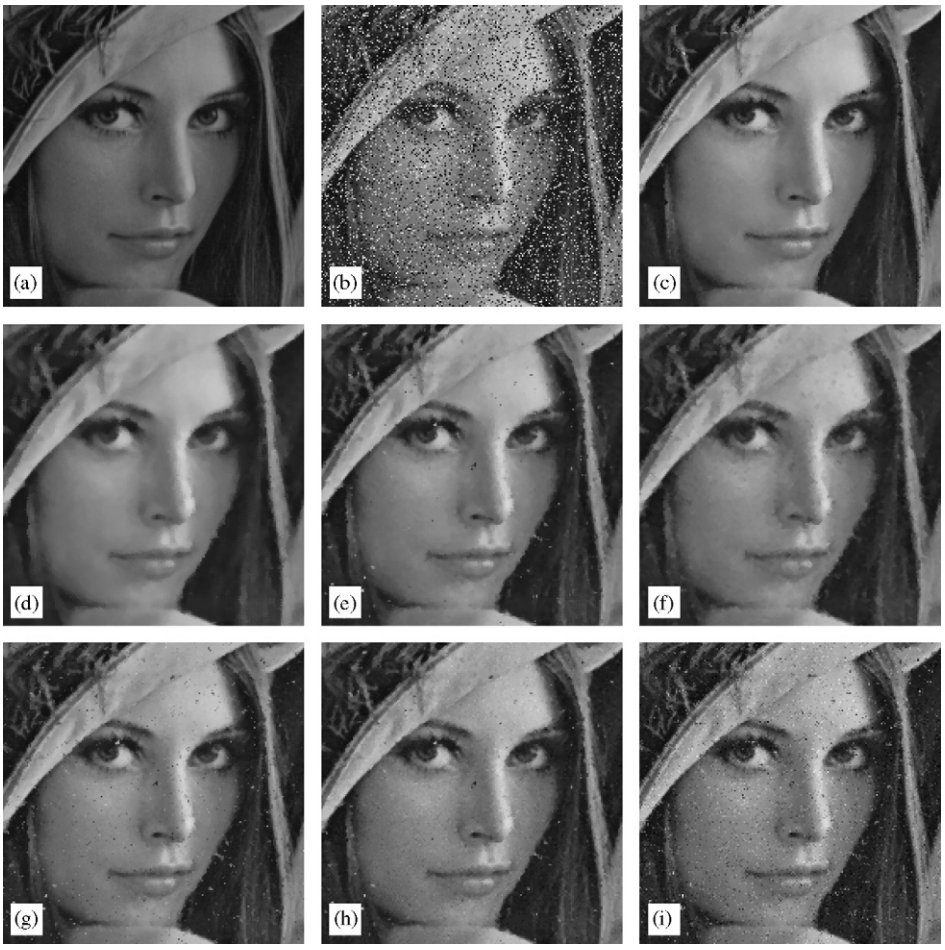

Fig. 6. (a) A part of the original Lena image. (b) The part corrupted with 25% random valued impulse noise (PSNR: 15.3 dB). (c) After the proposed FRINR method (PSNR: 33.9 dB). (d) After the CWM (PSNR: 30.8 dB). (e) After the GMED (PSNR: 30.0 dB). (f) After the IFCF (PSNR: 30.0 dB). (g) After the AFSF (PSNR: 29.3 dB). (h) After the FMF (PSNR: 29.9 dB). (i) After the FIRE (PSNR: 25.5 dB).

the proposed method FRINR (Fig. 5(c)) removes the impulse noise very well while preserving the fine details of the image. Other methods (Fig. 5(e) GMED, Fig. 5(h) IFCF and Fig. 5(g) LUM) remove the noise well, but they also tend to make the image blurrier. On the other hand, the FMF and AFSF can preserve the image details, but some impulses still remain in the image, as seen from Fig. 5(i) and 5(d).

Fig. 6 shows the restoration results of different filters for the "Lena" image. Fig. 6(a) and (b) show a magnified part of the original image and the image part corrupted with 25% random valued impulse noise. It is clearly seen from Fig. 6(c) that the proposed FRINR method received the best visual results which confirms the numerical results very well. We also observe that most of the filters do not remove all impulses (Fig. 6(e)–(i)) or destroy some image details making the image a bit more blurrier (Fig. 6(d) and 6(f)).

Finally, we also have investigated the time complexity of our method in comparison to the current filters. Time complexity refers to a function describing how much time it will take an algorithm to execute, based on the parameters of its input (in our case we have used NM pixels). The exact value of this function is usually ignored in favour of its order, expressed in the so-called Big-O notation (this is based on the limit of the time complexity function as the values of its parameters increase without bound). All the algorithms described here (including the proposed method) have a linear time complexity (i.e. $O(NM)$). This means that if the amount of pixels are multiplied by two (i.e. $2NM$) then the execution time will be twice as long (i.e. $2O(NM)$). This is also illustrated in Fig. 7 were we have plotted the mean execution time (vertical axis) of several methods for several images of different size (horizontal axis). These two

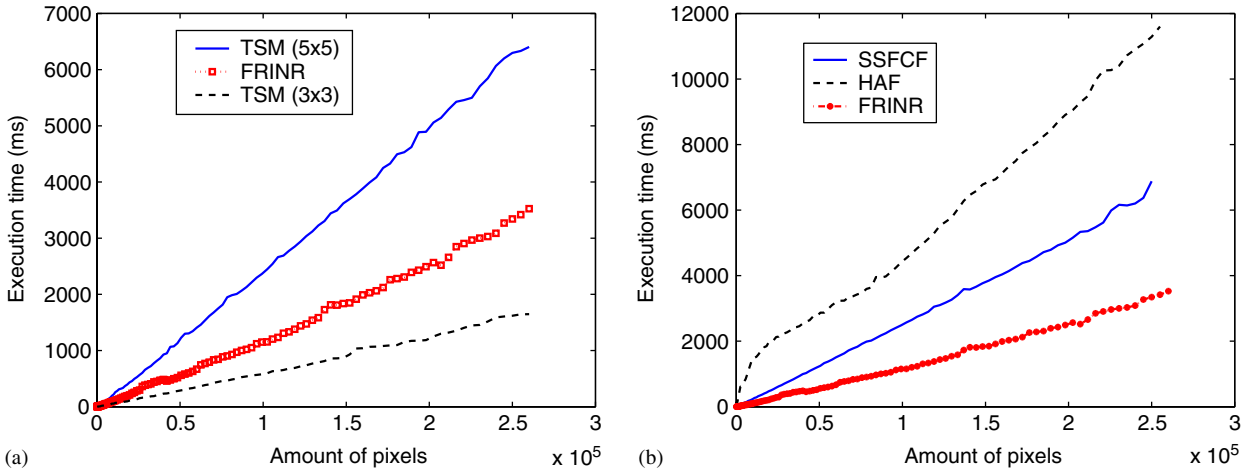

Fig. 7. The comparison of the mean execution time for (a) the comparison between the TSM method with window size 3×3 and 5×5 and the FRINRM; (b) the comparison between SSFCF, HAF and FRINRM.

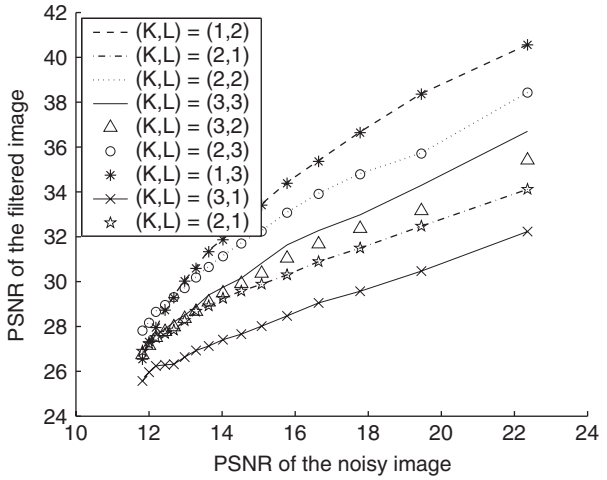

Fig. 8. The comparison of the mean execution time for different combinations of window size (K, L).

figures give an idea of the execution time of our proposed method in comparison to other state-of-the-art methods. In Fig. 7(a) we observe that the execution time of the FRINR method is situated between the execution time of TSM with window size 3×3 and TSM with window size 5×5. In Fig. 7(b) we have shown that the proposed method is faster than the two fuzzy based methods SSFCF and HAF.

In Fig. 8 we illustrated the influence of the window size. Several combinations of window sizes (K, L) were compared with each other. The horizontal axes of this figure show the PSNR value of the input noisy image and the vertical axes show the PSNR value of the filtered image using the FRINR method with several window sizes. If the noisy image has a PSNR value above 13 dB (corresponds with an impulse noise level lower than 60%) then we observe that the two window sizes $(K, L) = (1, 2)$ and $(K, L = 1, 3)$ clearly outperform other combinations of window sizes. If the input image were corrupted with more noise so that the PSNR of this image is lower than 12 dB then we observe that the window sizes $(K, L) = (2, 3)$ and $(K, L) = (2, 2)$ slightly outperform the other ones. Based on these results together with the time complexity analysis we can conclude that the best window size is $(K, L) = (1, 2)$, which corresponds with a 3×3 and 5×5 neighbourhood, respectively.

5. Conclusion

A new fuzzy random impulse noise reduction method (FRINR), which consists of two fuzzy detection methods and a fuzzy filtering algorithm, has been presented in this paper. This filter is especially developed for reducing all kinds of random valued impulse noise (which is more realistic than the salt and pepper noise). Its main advantage is that it removes impulse noise very well while preserving the fine image structures. Additionally, we have constructed the filter so that no parameters have to be chosen by the user. Experimental results have shown the feasibility of the new filter. A numerical measure, such as the PSNR, and visual observations (Fig. 5 and 6) have shown convincing results for grayscale images.

Acknowledgement

This work was financially supported by the GOA-project 12.0515.03 of Ghent University.

References

[1] K. Arakawa, Fuzzy rule-based image processing with optimization, in: E.E. Kerre, M. Nachtegael (Eds.), Fuzzy Techniques in Image Processing, Springer, New York, 2000, pp. 222–247.
[2] A. Ben Hamza, H. Krim, Image denoising: a nonlinear robust statistical approach, IEEE Trans. Signal Process. 49 (2001) 3045–3053.
[3] S. Bothorel, B. Bouchon, S. Muller, A fuzzy logic-based approach for semiological analysis of microcalcification in mammographic images, Internat. J. Intell. Systems 12 (1997) 819–843.
[4] T. Chen, K.K. Ma, L.H. Chen, Tri-state median filter for image denoising, IEEE Trans. Image Process. 8 (1999) 1834–1838.
[5] C. Cornelis, G. Deschrijver, E.E. Kerre, Classification of intuitionistic fuzzy implicators: an algebraic approach, in: Proc. 6th Joint Conf. on Information Sciences, 2002, pp. 105–108.
[6] F. Farbiz, M.B. Menhaj, A fuzzy logic control based approach for image filtering, in: E.E. Kerre, M. Nachtegael (Eds.), Fuzzy Techniques in Image Processing, Springer, New York, 2002, pp. 194–221.
[7] F. Farbiz, M.B. Menhaj, S.A. Motamedi, Edge preserving image filtering based on fuzzy logic, in: Proc. 6th EUFIT Conf. 1998, pp. 1417–1421.
[8] S.M. Guo, C.S. Lee, C.Y. Hsu, An intelligent image agent based on soft-computing techniques for color image processing, Expert Systems with Appl. 28 (2005) 483–494.
[9] R.C. Hardie, C.G. Boncelet, LUM filters: a class of rank-order-based filters for smoothing and sharpening, IEEE Trans. Signal Process. 41 (1993) 1061–1076.
[10] I. Kalaykov, G. Tolt, Real-time image noise cancellation based on fuzzy similarity, in: M. Nachtegael, D. Van der Weken, D. Van De Ville, E.E. Kerre (Eds.), Fuzzy Filters for Image Processing, Springer, Heidelberg, 2003, pp. 54–71.
[11] E.E. Kerre, Fuzzy Sets and Approximate Reasoning, Xian Jiaotong University Press, 1998.
[12] S.J. Ko, Y.H. Lee, Center weighted median filters and their applications to image enhancement, IEEE Trans. Circuits and Systems 38 (1991) 984–993.
[13] E.E. Kuruoğlu, C. Molina, S.J. Godsill, W.J. Fitzgerald, A new analytic representation for the α-stable probability density function, in: The Fifth World Meeting of the International Society for Bayesian Analysis (ISBA), Istanbul, August 1997.
[14] H.K. Kwan, Fuzzy filters for noise reduction in images, in: M. Nachtegael, D. Van der Weken, D. Van De Ville, E.E. Kerre (Eds.), Fuzzy Filters for Image Processing, Springer, Heidelberg, 2003, pp. 25–53.
[15] H.K. Kwan, Y. Cai, Fuzzy filters for image filtering, in: Proc. of Circuits and Systems, MWSCAS-2002, The 2002 45th Midwest Symposium, vol. III-672-5, 2002.
[16] C.S. Lee, Y.H. Kuo, Adaptive fuzzy filter and its application to image enhancement, in: E.E. Kerre, M. Nachtegael (Eds.), Fuzzy Techniques in Image Processing, Springer, New York, 2000, pp. 172–193.
[17] C.S. Lee, Y.H. Kuo, P.T. Yu, Weighted fuzzy mean filters for image processing, Fuzzy Sets and Systems 89 (1997) 157–180.
[18] M. Nachtegael, E.E. Kerre, Decomposing and constructing fuzzy morphological operations over alpha-cuts: continuous and discrete case, IEEE Trans. Fuzzy Systems 8 (2000) 615–626.
[19] M. Nachtegael, E.E. Kerre, Connections between binary, gray-scale and fuzzy mathematical morphologies, Fuzzy Sets and Systems 124 (2001) 73–85.
[20] B. Reusch, M. Fathi, L. Hildebrand, Fuzzy color processing for quality improvement, in: International Forum on Multimedia and Image Processing, 1998, pp. 841–848.
[21] F. Russo, Fire operators for image processing, Fuzzy Sets and Systems 103 (2) (1999) 265–275.
[22] F. Russo, G. Ramponi, A noise smoother using cascaded FIRE filters, in: Proc. FUZZ-IEEE'95-4th IEE Internat. Conf. on Fuzzy Systems, 1995, pp. 351–358.
[23] F. Russo, G. Ramponi, A fuzzy operator for the enhancement of blurred and noisy images, IEEE Trans. Image Process. 4 (1995) 1169–1174.
[24] F. Russo, G. Ramponi, A fuzzy filter for images corrupted by impulse noise, IEEE Signal Process. Lett. 3 (1996) 168–170.
[25] S. Schulte, M. Nachtegael, V. De Witte, D. Van der Weken, E.E. Kerre, A new two step color filter for impulse noise, in: Proc. East West Fuzzy Colloquium 2004, vol. 81, 2004, pp. 185–192.
[26] S. Schulte, M. Nachtegael, V. De Witte, D. Van der Weken, E.E. Kerre, A fuzzy impulse noise detection and reduction method, IEEE Trans. Image Process. 15 (5) (2006) 1153–1162.

[27] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, IEEE Trans. Systems Man and Cybernetics 15 (1) (1985) 116–132.
[28] D. Van De Ville, M. Nachtegael, D. Van der Weken, E.E. Kerre, W. Philips, Noise reduction by fuzzy image filtering, IEEE Trans. Fuzzy Systems 11 (4) (2001) 429–436.
[29] D. Van de Ville, W. Philips, I. Lemahieu, Fuzzy-based motion detection and its application to de-interlacing, in: E.E. Kerre, M. Nachtegael (Eds.), Fuzzy Techniques in Image Processing, Springer, New York, 2000, pp. 337–369.
[30] J.H. Wang, H.C. Chiu, An adaptive fuzzy filter for restoring highly corrupted images by histogram estimation, Proc. Nat. Sci. Counc. ROC(A) 23 (1999) 630–643.
[31] J.H. Wang, W.J. Liu, L.D. Lin, An histogram-based fuzzy filter for image restoration, IEEE Trans. Systems Man and Cybernetics Part B Cybernetics 32 (2) (2002) 230–238.
[32] H. Xu, G. Zhu, H. Peng, D. Wang, Adaptive fuzzy switching filter for images corrupted by impulse noise, Pattern Recognition Lett. 25 (2004) 1657–1663.