Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT•

Signal Processing 83 (2003) 1309–1342

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

On the geodesic paths approach to color image filtering

M. Szczepanski[a], B. Smolka[a,1], K.N. Plataniotis[b,*], A.N. Venetsanopoulos[b]

[a] Department of Automatic Control, Silesian University of Technology, Akademicka 16 Str, 44-100 Gliwice, Poland
[b] Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ont., Canada M5S 3G4

Received 6 December 2001; received in revised form 2 January 2003

Abstract

In this paper a novel method of noise reduction in color images is presented. The class of filters introduced here utilizes fuzzy membership functions defined over vectorial inputs connected via digital geodesic paths. The efficiency of the new filters is compared under a variety of performance criteria with the commonly used filters, such as the vector median and the generalized vector directional filter. It is shown that, compared to existing techniques, the filters introduced here are better able to suppress impulsive, Gaussian as well as mixed-type noise. Furthermore, the computational analysis included in this work shows that some members of the new filter family are computationally less demanding than the vector median filter.

Keywords: Color image processing; Noise reduction; Geodesic paths; Random walks; Fuzzy filters; Rank-typed filter; Impulsive noise; Computational complexity

1. Introduction

Image noise reduction without structure degradation is perhaps the most important low-level image processing task [15,21]. Several techniques have been proposed over the years. Among them are linear processing methods, whose mathematical simplicity and the existence of unifying theory make their design and implementation easy and attractive. However, not all filtering problems can be efficiently solved by using linear techniques. For example, conventional linear techniques cannot cope with nonlinearities of the image formation model and fail to preserve edges and image details.

To this end, nonlinear image processing techniques are introduced. Nonlinear techniques are able to suppress non-Gaussian noise and preserve important image elements, such as edges and details, and eliminate degradations occurring during image formation or transmission through nonlinear channels.

One of the most popular families of nonlinear filters for noise removal is the order-statistics filters family [7,11,14,15,21,30]. Their theoretical framework is based on the robust statistics as these filters utilize algebraic ordering of a windowed set to compute the output signal.

Let $\mathbf{F}(\mathbf{x})$ be a multichannel image and let W be a window of finite size k (filter length). The noisy image vectors inside the filtering window W are denoted as $\mathbf{F}_j$, $j = 0, 1, \ldots, k-1$. If the distance between two

* Corresponding author. Tel.: +1-416-946-5605; fax: +1-416-978-4425.
E-mail addresses: mszczepa@ia.polsl.gliwice.pl (M. Szczepanski), kostas@dsp.toronto.edu (K.N. Plataniotis).
[1] Partially supported by KBN Grant 7 T11A 010 21.

0165-1684/03/$ - see front matter
doi:10.1016/S0165-1684(03)00058-6

Nomenclature

$\mathbb{R}$	real numbers set
Π	geodesic path
λ	point on the polygonal line
χ	dissimilarity measure along a digital path
μ	similarity function
ψ	normalized similarity function
ω	number of geodesic paths connecting two distinct points
Ω	number of all geodesic paths included in the window W which start from the window's center pixel
ρ	distance function
ρ^S	geodesic distance defined over the path in set S
$\mathscr{H}$	digital image lattice
$\mathscr{N}$	neighborhood relation between lattice points
α	cooling parameter in iterative filtering procedure
β	new filter design parameter
γ	regularization parameter for adaptive filter design
σ	standard deviation

vectors $\mathbf{F}_i, \mathbf{F}_j$ is denoted as $\rho(\mathbf{F}_i, \mathbf{F}_j)$ then the scalar quantity

$$R_i = \sum_{j=0}^{k-1} \rho(\mathbf{F}_i, \mathbf{F}_j) \tag{1}$$

is the aggregated distance associated with the noisy vector $\mathbf{F}_i$ inside the processing window. Assuming, that a reduced ordering of the R_i's

$$R_{(0)} \leqslant R_{(1)} \leqslant \cdots \leqslant R_{(\tau)} \leqslant \cdots \leqslant R_{(k-1)} \tag{2}$$

implies the same ordering of the corresponding vectors $\mathbf{F}_i$

$$\mathbf{F}_{(0)} \leqslant \mathbf{F}_{(1)} \leqslant \cdots \leqslant \mathbf{F}_{(\tau)} \leqslant \cdots \leqslant \mathbf{F}_{(k-1)}. \tag{3}$$

Nonlinear ranked-type multichannel filters define the vector $\mathbf{F}_{(0)}$ as the result of the filtering operation.

The best known member of the family is the so-called *vector median filter* (VMF). The definition of the multichannel median is a direct extension of the ordinary scalar median definition with the L_1 or L_2 norm utilized to order vectors according to their relative magnitude differences [1].

Within the framework of ranked-type nonlinear filters, the orientation difference between color vectors can also be used to remove vectors with atypical directions. The *basic vector directional filter* (BVDF) is a ranked order filter, similar to the VMF, which uses the angle between two color vectors as the distance criterion. This criterion is defined as the scalar measure

$$a(\mathbf{F}_i, \mathbf{F}_j) = \cos^{-1}\left(\frac{\mathbf{F}_i \cdot \mathbf{F}_j^{\mathrm{T}}}{|\mathbf{F}_i||\mathbf{F}_j|}\right) \quad \text{with}$$

$$A_i = \sum_{j=0}^{k-1} a(\mathbf{F}_i, \mathbf{F}_j), \tag{4}$$

which assigns the corresponding aggregated distance to the noisy vector $\mathbf{F}_i$ inside the processing window W. As in the case of vector median filter, an ordering of the A_i's

$$A_{(0)} \leqslant A_{(1)} \leqslant \cdots \leqslant A_{(\tau)} \leqslant \cdots \leqslant A_{(k-1)} \tag{5}$$

implies the same ordering of the corresponding vectors $\mathbf{F}_i$

$$\mathbf{F}_{(0)} \leqslant \mathbf{F}_{(1)} \leqslant \cdots \leqslant \mathbf{F}_{(\tau)} \leqslant \cdots \leqslant \mathbf{F}_{(k-1)}. \tag{6}$$

The BVDF outputs the vector $\mathbf{F}_{(0)}$ that minimizes the sum of angles with all the other vectors within the processing window. Since the BVDF uses only information about vector directions (chromaticity information) it cannot remove achromatic noisy pixels from the image. To overcome the deficiencies of the BVDF, the *generalized vector directional filter* (GVDF) was introduced [29]. The GVDF generalizes BVDF in the sense that its output is a superset of the single BVDF output. The first vector in (6) constitutes the output of the BVDF, whereas the first τ vectors constitute the output of the GVDF. In this way

$$\mathrm{BVDF}\{\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{k-1}\} = \mathbf{F}_0, \tag{7}$$

$$\mathrm{GVDF}\{\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{k-1}\} = \{\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_\tau\}, \quad 1 \leqslant \tau \leqslant k-1. \tag{8}$$

The output of GVDF is subsequently passed through an additional filter in order to produce a single output vector. In this step the designer may only consider the magnitudes of the vectors $\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_\tau$ since they have approximately the same direction in the vector space. As a result, the GVDF separates the processing of color vectors into directional processing and then magnitude processing (the vector's direction signifies its chromaticity, while its magnitude is a measure of its brightness). The resulting cascade of filters is usually complex and the implementations may be slow since they operate in two steps [9,10].

To improve the efficiency of the directional filters, a new method called *directional-distance filter* (DDF) was proposed [7]. DDF constitutes a combination of VMF and BVDF and is derived by simultaneous minimization of the their defining functions. Another efficient rank-ordered operation called *hybrid directional filter* was proposed in [6]. This filter operates on the directional and the magnitude of the color vectors independently and then combines them to produce a final output. This hybrid filter, which can be viewed as a nonlinear combination of the VMF and BVDF filters, produces an output according to the following rule:

$$\mathbf{F}_{\mathrm{HyF}} = \begin{cases} \mathbf{F}_{\mathrm{VMF}} & \text{if } \mathbf{F}_{\mathrm{VMF}} = \mathbf{F}_{\mathrm{BVDF}}, \\ \left(\dfrac{\|\mathbf{F}_{\mathrm{VMF}}\|}{\|\mathbf{F}_{\mathrm{BVDF}}\|} \right) \cdot \mathbf{F}_{\mathrm{BVDF}} & \text{otherwise}, \end{cases} \tag{9}$$

where $\mathbf{F}_{\mathrm{BVDF}}$ is the output of the BVDF filter, $\mathbf{F}_{\mathrm{VMF}}$ is the output of the VMF and $\|\cdot\|$ denotes the norm of the vector.

All standard nonlinear filters, such as those briefly described here, are local operators working with a fixed supporting window W of finite length k that has in its center the pixel under consideration. Operation on the window involves examining the connections with other pixels. Ranked-type filters, such as the VMF calculate the aggregated distances amongst all elements in the window in order to determine the filters' output. However, such operations, which take place on a predefined supporting element, ignore the structural properties of the image resulting in oversmoothing and detail elimination. It is quite common during the application of such filters to have pixels grouped together at the same support window, although they belong to different semantic objects or are on different sides of edges. This results in blurring or complete masking of the actual image structures especially in the case of large window sizes.

In the literature, a number of methods have been introduced to prevent excessive smoothing. The most common approach is to restrict the size of the supporting window to ($k = 3 \times 3$) or ($k = 5 \times 5$). In this way, ranked-type nonlinear filters can remove additive Gaussian or impulsive noise and still preserve, relatively well, edges and other structural details in the image [21]. Another less common approach is to introduce heuristic modifications to the basic filtering structure by considering thresholds or pixel exclusions. Depending on predetermined parameters, a filtering structure is selectively applied to the same fixed supporting window [3,8,17].

For example, the so called *fast modified vector median filter* (FMVMF) introduced in [25,26], excludes the center pixel from the calculations of the aggregated distances associated with pixels from its neighborhood. This approach yields excellent performance for images corrupted with impulsive noise but is not intended to suppress Gaussian noise or mixed-type noise.

In this paper, we propose a different approach. Instead of using a fixed window or selectively apply the filtering procedure, we propose to exploit possible connections between successive image pixels using the concept of geodesic paths. According to the proposed here methodology, image pixels are grouped together forming paths that reveal the underlying structural dynamics of the image. Depending on the design principles and the computational constraints imposed on the design, our framework allows for paths to be considered on the entire image or to be restricted in a predefined search area. In this work, we focus on the latter case. To facilitate comparison with existing ranked-type operations and to illustrate the computational efficiency of the proposed framework, we allow the path searching area to match the window W used by the ranked-type filters. However, instead of the indiscriminately use of the window pixels, an approach advocated by all existing multichannel filters, the proposed here framework allows for the formation of a number of geodesic paths which in turn are used to determine the weights of a weighted average type of filtering operation.

The path displacements evaluated over all possible geodesic paths, are used to derive fuzzy membership functions that quantify similarity between vectorial inputs. The proposed filtering structure is then using the function outputs to appropriately weight input contributions in order to determine the filter result. The proposed filtering schemes parallelize the familiar structure of the adaptive multichannel filter introduced in [18] and they can successfully eliminate Gaussian, impulsive as well as mixed-type additive noise. However, thanks to the introduction of the geodesic paths in its supporting element, the new filters not only preserve edges and fine image details, but even enhance them, acting as an image sharpening operator.

This paper is organized as follows. In Section 2 the general concept of the geodesic paths is introduced and its application to the problem of the noise suppression is briefly discussed. In Section 3, the new filtering framework is presented. The motivation and design characteristics are discussed in details there and two different design approaches are analyzed. The complexity of the proposed filters and related optimization issues are discussed in Section 4, while Section 5 presents simulation results of experiments performed on artificial test images as well as on natural color images. Comparisons, in terms of image restoration performance, with commonly used multichannel filters are reported there. Finally, Section 6 summarizes this paper.

2. Geodesic paths approach

Let us assume, that $\mathbb{R}^2$ is the Euclidean space, S is a planar subset of $\mathbb{R}^2$ ($S \subset \mathbb{R}^2$) and x, y are points belonging to set S. A path from x to y is a continuous mapping $\Pi : [a,b] \to S$, such that $\Pi(a) = x$ and $\Pi(b) = y$. The point x is considered as the starting point while y is the ending point on the path Π [4,5].

An increasing polygonal line P on the path Π is any polygonal line such that $P = \{\Pi(\lambda_i)\}_{i=0}^{n}$, $a = \lambda_0 < \cdots < \lambda_n = b$. The length of the polygonal line P is considered to be the total sum of its constitutive line segments $L(P) = \sum_{i=1}^{n} \rho(\Pi(\lambda_{i-1}), \Pi(\lambda_i))$, where $\rho(x,y)$ is the distance between the points x and y, when a specific metric is adopted. A path Π from x to y is called rectifiable, if and only if $L(P)$, where P is an increasing polygonal line, is bounded. Its upper bound is called the length of the path Π.

The geodesic distance $\rho^S(x,y)$ between points x and y is the lower bound of the length of all paths leading from x to y which are totally included in S. If such paths do not exist, then the value of the geodesic distance is set to ∞. In general $\rho^S(x,y) \geqslant \rho(x,y)$. However, if the set S is convex, meaning that there are no points on the line between x and y that are not members of S, the geodesic distance verifies $\rho^S(x,y) = \rho(x,y)$.

The notion of a path can be extended to a lattice, which is a set of discrete points on the plane, in our case the spatial locations of the image pixels. Let a digital lattice $\mathscr{H} = (\mathbf{F}, \mathscr{N})$ be defined by $\mathbf{F}$, which is the set of all points of the plane (pixels of a color image) and a neighborhood relation $\mathscr{N}$ between the lattice points [22,23]. In the case of the ranked-type nonlinear filters the processing window W forms a lattice where $\mathscr{N}$ is defined through the window size.

A digital path $P = \{p_i\}_{i=0}^{n}$ defined on the lattice $\mathscr{H}$ is a sequence of neighboring points $(p_{i-1}, p_i) \in \mathscr{N}$. The length $L(P)$ of the digital path P is simply $\sum_{i=1}^{n} \rho^{\mathscr{H}}(p_{i-1}, p_i)$, where $\rho^{\mathscr{H}}$ denotes the distance between two neighboring points of the lattice $\mathscr{H}$.

Constraining the paths to be totally included in a predefined set $W \in \mathbf{F}$ yields the digital geodesic distance ρ^W. An 8-neighborhood system is considered in this work with a topological distance of 1 assigned between two neighboring points. In this case the set W is simply the well-known supporting window used in ranked-type filters. All paths considered here are included in the neighborhood W (Fig. 1).

Two pixels which are located at spatial coordinates (i,j) and (k,l) are called connected (hereafter denoted as $(i,j) \Leftrightarrow (k,l)$), if there exists a geodesic path $P^W\{(i,j),(k,l)\}$ contained in the set W starting from (i,j) and ending at (k,l).

If two pixels (i_0,j_0) and (i_n,j_n) are connected by a geodesic path $P_m^W\{(i_0,j_0),(i_1,j_1),\ldots,(i_n,j_n)\}$ of length n then $\chi_m^{W,n}$

$$\chi_m^{W,n}\{(i_0,j_0),(i_n,j_n)\} = \sum_{k=0}^{n-1} \|\mathbf{F}(i_{k+1},j_{k+1}) - \mathbf{F}(i_k,j_k)\|, \tag{10}$$

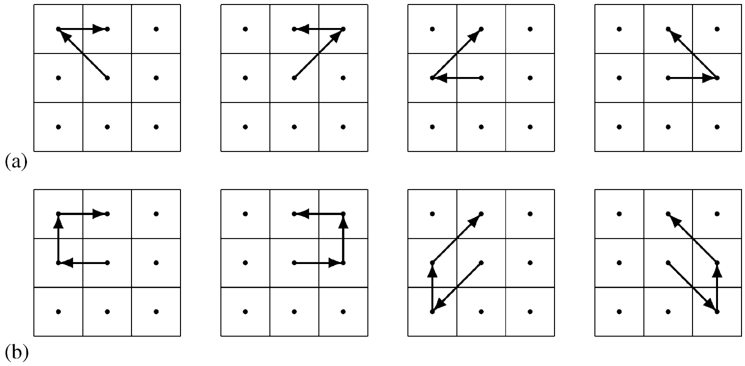


Fig. 1. Geodesic paths of finite length: (a) $n=2$, (b) $n=3$, connecting two neighboring points within a predefined window W when the 8-neighborhood system is applied.

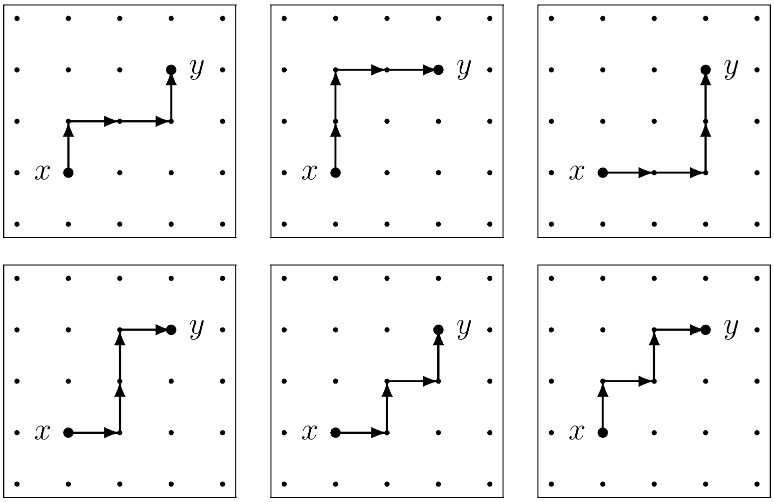


Fig. 2. There are six paths of length 4 connecting point x and y when the 4-neighborhood system is used.

where m is the path index, is a measure of dissimilarity between pixels (i_0, j_0) and (i_n, j_n), along a specific geodesic path P_m^W joining (i_0, j_0) and (i_n, j_n) [5,28]. If a path joining two distinct points x, y, such that $\mathbf{F}(x) = \mathbf{F}(y)$ consists of lattice points of the same channel values, then $\chi^{W,n}(x, y) = 0$ otherwise $\chi^{W,n}(x, y) > 0$.

In general, two distinct pixel's locations on the image lattice can be connected by many paths. Moreover the number of possible geodesic paths of certain length n connecting two distinct points depends on their locations, length of the path and the neighborhood system used (Figs. 1 and 2).

Let us now define a similarity function, analogous to a membership function used in fuzzy systems, between two pixels connected through all possible geodesic digital paths leading from (i, j) to (k, l) as follows:

$$\mu^{W,n}\{(i,j),(k,l)\} = \sum_{m=1}^{\omega} g(\chi_m^{W,n}\{(i,j),(k,l)\}), \quad (11)$$

where ω is the number of all paths connecting (i, j) and (k, l), $\chi_m^{W,n}\{(i,j),(k,l)\}$ is a dissimilarity value along a specific path m from the set of all ω possible paths leading from (i, j) to (k, l) and $g(\cdot)$ is a smooth function of $\chi_m^{W,n}$. By definition $\mu^{W,n}\{(i,j),(k,l)\}$ returns a value evaluated over all possible routes linking the starting point (i,j) to the endpoint (k,l).

The smooth function $g:(0;\infty]\rightarrow\mathbb{R}$ should satisfy the following conditions:

(1) g is a decreasing function in $(0;\infty]$,
(2) g is convex in $(0;\infty]$,
(3) $g(0)=1$,
(4) $g(\chi)\rightarrow 0$, when $\chi\rightarrow\infty$.

Several functions satisfying the above conditions have been proposed in the literature [13,21,24]

$$g_0(x)=\mathrm{e}^{-\beta_1 x^2},\qquad \beta_0\in(0;\infty), \tag{12}$$

$$g_1(x)=\mathrm{e}^{-\beta_1 x},\qquad \beta_1\in(0;\infty), \tag{13}$$

$$g_2(x)=\frac{1}{1+\beta_2 x},\qquad \beta_2\in(0;\infty), \tag{14}$$

$$g_3(x)=\frac{1}{(1+x)^{\beta_3}},\qquad \beta_3\in(0;\infty), \tag{15}$$

$$g_4(x)=1-\frac{2}{\pi}\arctan(\beta_4 x),\qquad \beta_4\in(0;\infty), \tag{16}$$

$$g_5(x)=\frac{2}{1+\mathrm{e}^{\beta_5 x}},\qquad \beta_5\in(0;\infty), \tag{17}$$

$$g_6(x)=\frac{1}{1+x^{\beta_6}},\qquad \beta_6\in(0;1), \tag{18}$$

$$g_7(x)=\begin{cases}1-\beta_7 x & \text{if } x<1/\beta_7,\\ 0 & \text{if } x\geqslant 1/\beta_7,\end{cases}\qquad \beta_7\in(0;\infty). \tag{19}$$

In this work the exponential function of (13) is used, as it proved to yield very good results. Therefore,

$$\mu^{W,n}\{(i,j),(k,l)\}=\sum_{m=1}^{\omega}\exp[-\beta\chi_m^{W,n}\{(i,j),(k,l)\}], \tag{20}$$

where β is the filter design parameter.

For $n=1$ and a square (3×3) window W the similarity function μ is defined as follows:

$$\mu^{W,1}\{(i,j),(k,l)\}=\exp\{-\beta\|\mathbf{F}(i,j)-\mathbf{F}(k,l)\|\} \tag{21}$$

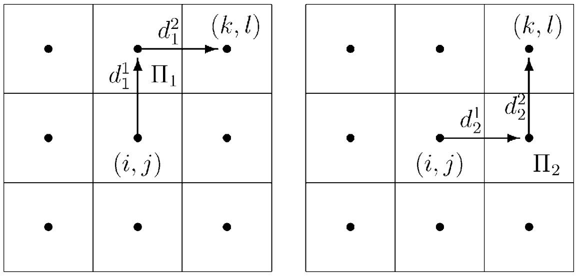


Fig. 3. Geodesic paths of length $n=2$ connecting points (i,j) and (k,l).

and then if $\mathbf{F}(i,j)=\mathbf{F}(k,l)$ then $\chi^{W,n}\{(i,j),(k,l)\}=0$, $\mu\{(i,j),(k,l)\}=1$, and for $\|\mathbf{F}(i,j)-\mathbf{F}(k,l)\|\rightarrow\infty$, then $\mu\rightarrow 0$ [17].

Fig. 3 illustrates the calculation of the similarity function between two points connected by two geodesic paths of length $n=2$. In this case

$$\chi_1^{W,2}\{(i,j),(k,l)\}=d_1^1+d_1^2,$$
$$\chi_2^{W,2}\{(i,j),(k,l)\}=d_2^1+d_2^2, \tag{22}$$

with d_1^1, d_1^2 distances between neighboring points on the path Π_1 defined according to (10), while d_2^1, d_2^2 are similarly defined on Π_2. The total similarity can be expressed as follows:

$$\mu^{W,2}=\exp(-\beta\chi_1^{W,2})+\exp(-\beta\chi_2^{W,2}). \tag{23}$$

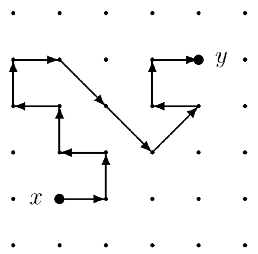
A normalized form of the similarity function can be defined as follows:

$$\psi^{W,n}\{(i,j),(k,l)\}=\frac{\mu^{W,n}\{(i,j),(k,l)\}}{\sum_{(l,m)\Leftrightarrow(i,j)}\mu^{W,n}\{(i,j),(l,m)\}}, \tag{24}$$

where $(l,m)\Leftrightarrow(i,j)$ denotes all points (l,m) connected by digital geodesic paths with (i,j). It is clearly seen from (24) that the normalized similarity function satisfies the following property:

$$\sum_{(k,l)\Leftrightarrow(i,j)}\psi^{W,n}\{(i,j),(k,l)\}=1. \tag{25}$$

Assuming that the pixel located at position (i,j) is the pixel under consideration, with $\mathbf{F}(k,l)$ representing



(a)

1	2	3	2	1
2	2	4	2	2
3	4	•	4	3
2	2	4	2	2
1	2	3	2	1

(b)

Fig. 4. (a) The SAP on the two dimensional lattice (b) and the number of possible paths of length $n=2$ connecting center point with its neighbors in W.

the (k,l) pixel included in the supporting element W which is connected to (i,j) via a geodesic path, the filtering result $\hat{\mathbf{F}}(i,j)$ is given as follows:

$$\hat{\mathbf{F}}(i,j) = \sum_{(k,l)\Leftrightarrow(i,j)} \psi^{W,n}\{(i,j),(k,l)\} \cdot \mathbf{F}(k,l). \tag{26}$$

As can be easily noticed, $\hat{\mathbf{F}}$ is the weighted average of all points connected by geodesic paths to the central pixel (i,j).

3. Geodesic path based filters

3.1. Digital path concept

The performance of the new filters strongly depends on the type of digital paths selected. Different models of paths result in application-specific filters, which are able to suppress certain types of noise. In this paper we concentrate on the *self-avoiding path model* (SAP), which provides a model suitable for image processing applications [27].

The SAP is a special type of path taken along the image lattice so that adjacent pairs of edges in the sequence share a common vertex of the lattice. In the SAP approach no vertex is visited more than once resulting in a trajectory that never intersects itself. In other words the SAP is a path that does not pass through the same lattice point twice (Fig. 4a). Given the fact that on the two-dimensional lattice, a digital path is a finite sequence of distinct lattice points $(i_0,j_0),(i_1,j_1),\ldots,(i_n,j_n)$, which are in a neighborhood relation $\mathcal{N}$, the (SAP) should satisfy the following condition:

$$\forall_{k\neq m} (i_k,j_k) \neq (i_m,j_m). \tag{27}$$

Based on the concept of SAPs a family of image processing filters will be introduced.

The first member of the family, hereafter denoted as SAP filter, allows for the formulation of paths of size n, where n is the fixed length of the digital path. Given a path of length n, a supporting window W of size $(k\times k)$, with $k=2n+1$, is considered with the point under consideration in its center. It should be emphasized that the number of possible paths ω leading from the center pixel to its neighbors inside the window W depends on the particular location of the neighborhood pixel. In this way for filters based on paths of length 1, 2, 3 processing window of size (3×3), (5×5), (7×7), respectively, will be used. This is illustrated in Fig. 4b when path length $n=2$ defines the size of processing window W. In this case weights used in (26) associated with points located far from the center are relatively small.

The computational complexity of the SAP filter depends on the path length n and the number of paths, which can be constructed in the supporting window W of size $(k\times k)$. It is not hard to see that for large k, which may be the case in certain applications, the computational complexity of the filter renders it inapplicable. To decrease the computational burden, another member of the family is introduced. In this new algorithm, called *fast random walk approach* (FRWA), the size of the supporting window (mask) W is set to (3×3) independently of the geodesic path's length (Fig. 1).

3.2. Iterative behavior of the new filter class

The parameter β in (13) regulates the smoothness of the similarity function. Since the filtering structure of (26) is a regression estimator, which provides a smooth interpolation among the observed, noise-corrupted image vectors, the parameter β provides the required balance between smoothing and detail preservation. Therefore, it is not surprising that the best results are obtained when the smoothing operator $\hat{\mathbf{F}}$ in (26) is applied in an iterative way.

Starting with a low value of β enables the smoothing of the image noise components. At each iteration step the parameter β is increased. In particular, β can be modified as follows:

$$\beta(\kappa) = \beta(\kappa - 1)\alpha, \quad \kappa = 1, 2, \ldots, \tag{28}$$

where κ is the iteration number.

However, in this case two parameters α and β are needed to regulate the performance of the filter. In order to make the new filter less dependent on the initial parameter values, an adaptively determined β was introduced. The estimation of β is based on the assumption that in noisy images sample pixels values are varying heavily. Therefore, some measure of dispersion of the pixel values should be used for the calculation of β.

In this paper the parameter β is determined adaptively from the noise corrupted image data available through the fixed filter processing window W. Since the parameter β is by construction inversely proportional to the generalized standard deviation of samples in W, its estimated value is obtained as follows:

$$\hat{\beta} = \gamma \left(\frac{1}{NL} \sum_{i,j \in W} \sum_{l=1}^{L} (F_l(i,j) - \bar{F}_l)^2 \right)^{-1/2} \tag{29}$$

where N is the number of pixels in the processing window W, L is the dimensionality of the image vectors (in the RGB color space $L = 3$), $\bar{F}_l$ denotes the mean value of the lth component in window W and γ is a normalizing parameter.

The only free parameter in (29) is the design parameter γ. This parameter adds an extra degree of control over the detail preservation nature of the filter. For example using small γ values we obtain flat, more homogeneous images while increasing this value preserves sharp edges and fine details. Experimentation with a wide range of synthetic and natural images revealed that the best results are obtained for values of $\gamma \in [4, 10]$ while a value of $\gamma = 6$ results in good performance regardless of the specifics of the input image.

4. Computational complexity and fast filter design

Apart from the numerical behavior of any proposed algorithm, its computational complexity is a realistic measure of its practicality and usefulness, since it determines the required computing power and processing (execution) time. A general framework to evaluate the computational requirements of image filtering algorithms based on fixed processing window is given in [2,20]. The requirement of this approach is that the filter window W is symmetric $(k \times k)$ and contains k^2 vector samples of dimension L. In most image processing applications a value $k = 3$ is considered, while for color RGB images $L = 3$.

The computational complexity of a specific filter is given in terms of the total execution time needed for a complete filtering cycle. The total time T is calculated as

$$T = \sum_k \upsilon_k \vartheta_k, \tag{30}$$

where ϑ_k is the number of particular operations required for a complete cycle, and υ_k is the relative weight of this operation.

In our analysis the following operations are used: ADDS (additions), MULTS (multiplications), DIVS (divisions), SQRTS (square roots), COMPS (comparisons), ARCCOS (arc cosines) and EXPS (exponents). The determination of the weights υ_k of different operations in (30) is beyond the scope of this work, (we assume $\upsilon_k = 1$).

Since the structure of the new filters is not based on fixed window the methodology presented in [2,20] cannot be directly applied to evaluate the new filters' complexity. The complexity of the proposed here filters depends mostly on the number of possible geodesic paths, which in turn depends on the path type and its length.

For the general SAP filter the number of all possible geodesic paths is application dependent. However for given path length n, the number of geodesic paths Ω can be evaluated experimentally (there is no rigorous mathematical theory of self-avoiding walks [12]) and thus filters with predefined path length are considered here in order to facilitate the comparison with the existing rank-ordered nonlinear filters. In the sequence it is assumed that Ω is a number of all possible paths, n is the path length and L is the vector space dimension, with Table 1 depicting the number of possible paths corresponding to the SAP and FRWA filters considered in this comparison.

Table 1
Number of possible geodesic paths Ω in dependence on path length n

n	1	2	3	4
SAP	8	56	368	2336
FRWA	8	24	56	69

Based on the above assumptions the complexity of the SAP and FRWA filters can be determined as follows:

(1) Filtering of 1 pixel requires computation of all weights $\psi^{W,n}$ (see point 2), $L(\Omega - 1)$ additions and $L\Omega$ multiplications.
(2) Computation of all weights $\psi^{W,n}$ requires computation of all similarity functions $\mu^{W,n}$ (see point 3), Ω divisions and $(\Omega - 1)$ additions.
(3) Computation of all similarity functions $\mu^{W,n}$ requires Ω computations of distance $\chi_m^{W,n}$ (see point 4), $(\Omega - 1)$ additions, Ω multiplications and Ω computations of an exponent.
(4) Computation of one distance $\chi_m^{W,n}$ along path m requires n computations of Euclidean distance (if the L_2 metric is used) and $(n - 1)$ additions.
(5) Computation of one particular Euclidean distance requires L multiplications, $2L$ additions and 1 square root.

Thus the total number of operations needed to implement the filters is

$$
\begin{aligned}
&(2nL\Omega + \Omega p + L\Omega - L - 2)\mathrm{ADDS}\\
&+(\Omega + L\Omega + 2n)\mathrm{MULTS}\\
&+\Omega\,\mathrm{DIVS} + \Omega n\,\mathrm{SQRTS} + \Omega\,\mathrm{EXPS}.
\end{aligned}
$$

Using the framework in [2] and assuming that size of the processing window is $(k \times k)$ the computational complexity for the VMF, BVDF and DDF can be evaluated. Assuming that the L_2 norm is used, the number of basic operations required to calculate single VMF output is

$$
\begin{aligned}
&[(2L+3)k^3 - (L+2)k^2 - (L+1)k]\mathrm{ADDS}\\
&+L(k^3 - \tfrac{1}{2}k(k+1))\mathrm{MULTS}\\
&+(k^3 - \tfrac{1}{2}k(k+1))\mathrm{SQRTS}\\
&+(k^2 - 1)\mathrm{COMPS}.
\end{aligned}
$$

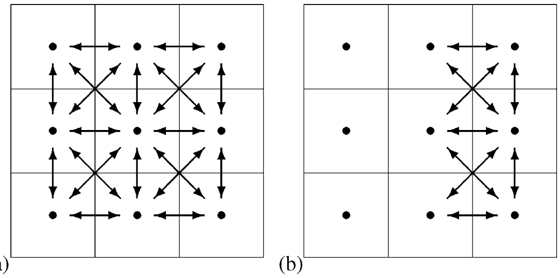
In the BVDF, instead of the L_2 norm, the angular distance is utilized, therefore

$$
\begin{aligned}
&[(5L+3)k^3 - (2.5L+2)k^2 - (2.5L+1)k]\mathrm{ADDS}\\
&+(3L+1)(k^3 - \tfrac{1}{2}k(k+1))\mathrm{MULTS}\\
&+(k^3 - \tfrac{1}{2}k(k+1))\mathrm{DIVS}\\
&+(k^3 - \tfrac{1}{2}k(k+1))\mathrm{SQRTS}\\
&+(k^3 - \tfrac{1}{2}k(k+1))\mathrm{ARCOS}\\
&+(k^2 - 1)\mathrm{COMPS}.
\end{aligned}
$$

Finally for the DDF filter which utilizes both the angular and the L_2 distance the total number of operations needed for a complete filtering cycle is

$$
\begin{aligned}
&[(7L+4)k^3 - (3.5L+1)k^2 - 3.5Lk]\mathrm{ADDS}\\
&+[(3L+1)(k^3 - \tfrac{1}{2}k(k+1)) + k^2]\mathrm{MULTS}\\
&+(k^3 - \tfrac{1}{2}k(k+1))\mathrm{DIVS}\\
&+2(k^3 - \tfrac{1}{2}k(k+1))\mathrm{SQRTS}\\
&+(k^3 - \tfrac{1}{2}k(k+1))\mathrm{ARCCOS} + (k^2 - 1)\mathrm{COMPS}.
\end{aligned}
$$

It should be emphasized at this point that the computational complexity analysis of the new filter is based on a straightforward application of the described algorithms without any consideration of a particular implementation. However, it is possible to reduce the



(a) (b)

Fig. 5. Illustration of the FRWA filter optimization: (a) all necessary distances to calculate, (b) new distances in processing window if last processed pixel was to the left.

computational complexity of the proposed filters. To illustrate this the FRWA filter is considered. The analysis of the filtering equation reveals that the L_2 distance should be evaluated n times for each path of length n. If the total number of paths in the supporting window is Ω, the number of L_2 norm evaluations is (Ωn). However, most of these calculations are unnecessary, since values already computed for other paths can be used. For example in a (3×3) window there are only 20 possible distances to be calculated (Fig. 5a). These values can be computed once and stored in a look-up table which can be used to determine the path related weights. Furthermore, other techniques used to improve the performance of the VMF presented in [2] can be applied in the SAP or FRWA filter design, (Fig. 5b).

Finally, it should be noted that the adaptive determination of the parameter β requires:

$$\boxed{\begin{array}{l}(k^4+2k^2L-3)\mathrm{ADDS}+2(k^2L+1)\mathrm{MULTS}\\ +1\mathrm{DIVS}+1\mathrm{SQRTS}\end{array}}$$

Table 2 summarizes the total number of operation for different filters, with SAP_n denoting the SAP algorithm with path of length n, FRWA_n denoting straightforward application of FRWA algorithms and FRWA_n^* the optimized version of FRWA. Finally $\hat{\beta}_{k\times k}$ is used to indicate an adaptive determination of β through (29) in a window W of size $k\times k$.

As it can be observed, the fast implementation of the proposed filter is computationally more attractive than the VMF and it significantly outperforms the filters based on angular distances.

5. Simulation results

In this section we evaluate the performance of the new class of filters and compare them with a number of image processing filters listed in Table 4 using a set

Table 2
Number of elementary operations for a complete processing cycle

	ADDS	MULTS	DIVS	SQRTS	EXPS	COMPS	ARCCOS
SAP_2	947	228	56	112	56	—	—
SAP_3	8827	1478	368	1104	368	—	—
FRWA_2	403	100	24	48	24	—	—
FRWA_3	1139	230	56	168	56	—	—
FRWA_2^*	169	22	24	9	24	—	—
FRWA_3^*	721	24	56	9	56	—	—
$\mathrm{VMF}_{3\times3}$	186	63	—	21	—	8	—
$\mathrm{VMF}_{5\times5}$	855	330	—	110	—	24	—
$\mathrm{BVDF}_{3\times3}$	375	210	21	21	—	8	21
$\mathrm{BVDF}_{5\times5}$	1970	1100	110	110	—	24	110
$\mathrm{DDF}_{3\times3}$	540	282	21	42	—	8	21
$\mathrm{DDF}_{5\times5}$	2785	1455	110	220	—	24	110
$\hat{\beta}_{3\times3}$	132	56	1	1	—	—	—
$\hat{\beta}_{5\times5}$	772	152	1	1	—	—	—

of synthetic and natural images corrupted by additive noise.

5.1. Noise model

In many practical applications images are corrupted by noise caused either by faulty image sensors or due to transmission corruption resulting from man-made phenomena such as ignition transients in the vicinity of the receivers or even natural phenomena such as lightning in the atmosphere. Transmission noise, also known as *salt & pepper* noise in gray-scale imaging, is modelled after an impulsive distribution. However, a common difficulty encountered in the studies of the effect of noise on image degradation is the lack of a commonly accepted multivariate impulsive noise model.

A number of simplified models has been introduced recently, to assist in the performance evaluation of the different color image filters. The impulsive noise model considered in this paper is as follows [21,31]:

$$\mathbf{F}_\mathrm{I} = \begin{cases} (F_1, F_2, F_3)^\mathrm{T} & \text{with probability } (1-p), \\ (d, F_2, F_3)^\mathrm{T} & \text{with probability } p_1 p, \\ (F_1, d, F_3)^\mathrm{T} & \text{with probability } p_2 p, \\ (F_1, F_2, d)^\mathrm{T} & \text{with probability } p_3 p, \\ (d, d, d)^\mathrm{T} & \text{with probability } p_4 p \end{cases} \tag{31}$$

with $\mathbf{F}_\mathrm{I}$ is the noisy signal, $\mathbf{F} = (F_1, F_2, F_3)^\mathrm{T}$ is the noise-free color vector, d is the impulse value and $\sum_{i=1}^{4} p_i = 1$.

Impulse d can have either positive or negative values. We further assume that $|d| \gg F_1, F_2, F_3$. Thus, when an impulse is added or subtracted, forcing the pixel value outside the [0,255] range, clipping is applied to move the corrupted noise value into the integer range specified by the 8-bit arithmetic.

In many practical situations an image is often corrupted by both additive Gaussian noise due to faulty sensors and impulsive transmission noise introduced by environmental interferences or faulty communication channels. An image can therefore be thought of as being corrupted by mixed noise according to the following model:

$$\mathbf{F}_\mathrm{M} = \begin{cases} \mathbf{F} + \mathbf{F}_\mathrm{G} & \text{with probability } (1 - p_\mathrm{I}), \\ \mathbf{F}_\mathrm{I} & \text{otherwise}, \end{cases} \tag{32}$$

where $\mathbf{F}$ is the noise-free color signal with the additive noise $\mathbf{F}_\mathrm{G}$ modelled as zero mean *white* Gaussian noise and $\mathbf{F}_\mathrm{I}$ transmission noise modelled as multivariate impulsive noise with $\mathbf{p}_\mathrm{I} = (p, p_1, p_2, p_3)$ determining the intensity and distribution of the impulsive noise contamination [21].

5.2. Application to artificial images

The use of nonlinear filters in color image processing is motivated primarily by the good performance of the filters near edges and other sharp signal transitions. Edges are basic images features which carry valuable information, useful in image analysis and object classification. Therefore, any nonlinear processing operator is required to preserve edges and smooth out noise without altering sharp signal transitions.

Simple examples are introduced in this section to illustrate the effectiveness of the proposed filtering operations near noisy edges. The *self-avoiding walk* (SAP) and the *fast random walk* (FRWA) algorithms are compared in terms of their performance with the VMF and the *arithmetic mean filter* (AMF). Predefined constant filter parameters were used in all experiments. The SAP and FRWA filters use paths of length 2 with $\beta = 20$ and $\alpha = 1.2$. The AMF and VMF operate on a filtering window of size (3×3). It should be pointed that those parameters used for the FRWA and SAP filters are not optimal and in the most cases better results could be obtained. Especially, optimal values for images corrupted with "pure" impulsive noise differ significantly, however in practical situations optimal values of design filter parameters are not known, and therefore we decided to use fixed values of these parameters.

To quantitatively evaluate the behavior of the algorithms, simple three-channel synthetic images were prepared. For simplicity results obtained for the green channel are presented.

To examine the performance of the filter in the case of an artificial edge, the synthetic test image "pyramid" was constructed. The three-channel image of size (90×90) contains a top-cut pyramid, which is used to emulate a "ramp-edge" scenario. Fig. 6a shows image

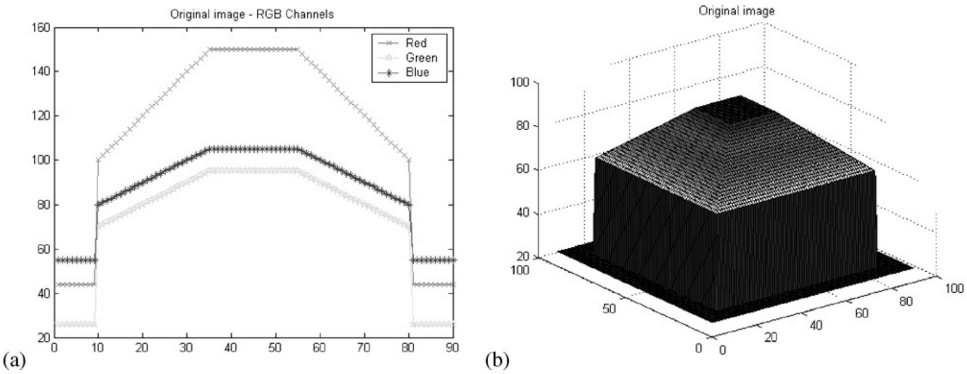


Fig. 6. Original noise-free pyramid image: (a) line plot of row 45, (b) 3D plot of green channel.

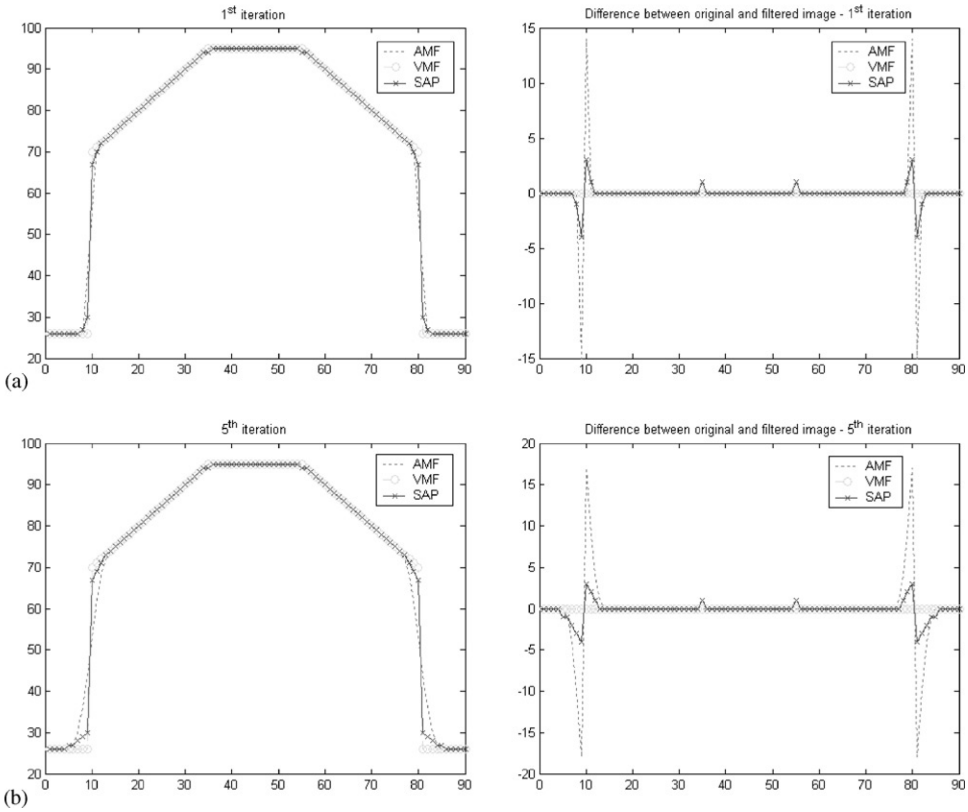


Fig. 7. Results of applying tested algorithms to the noise-free pyramid image: (a) first and (b) fifth iteration.

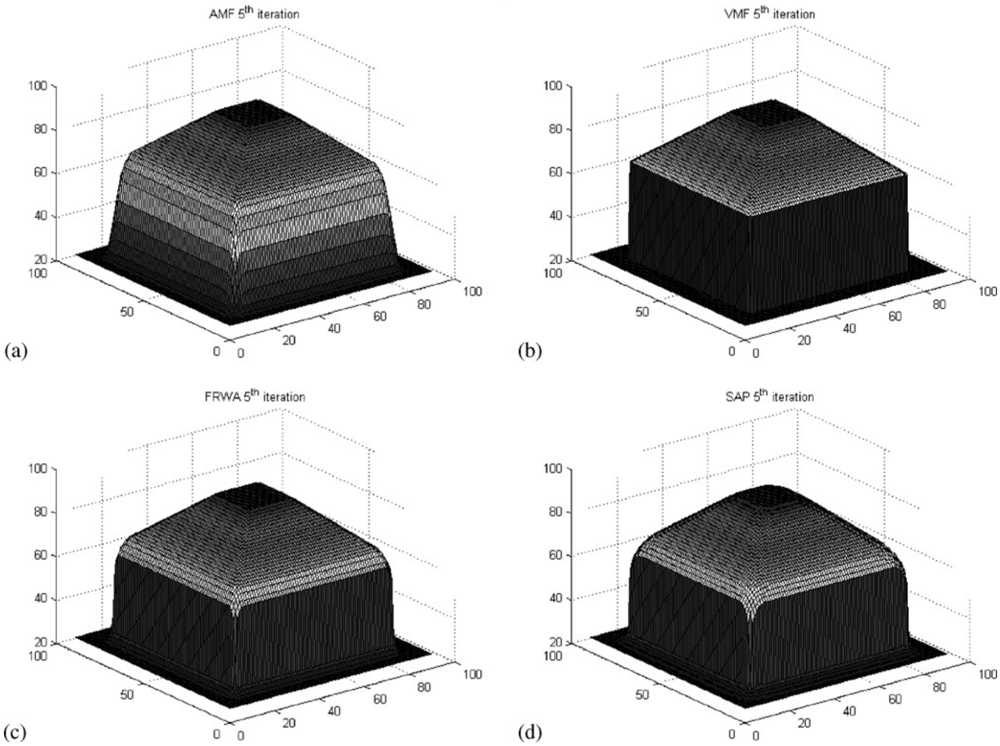


Fig. 8. 3D plots with results of applying filters to the noise-free pyramid image: (a) AMF, (b) VMF, (c) FRWA and (d) SAP, (five iterations).

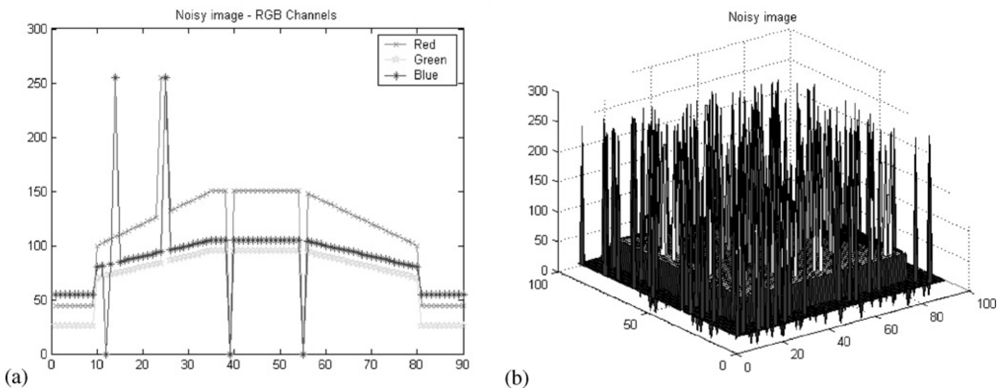


Fig. 9. Pyramid image corrupted by 10% impulsive noise: (a) line plot of row 45, (b) 3D plot of the green channel.

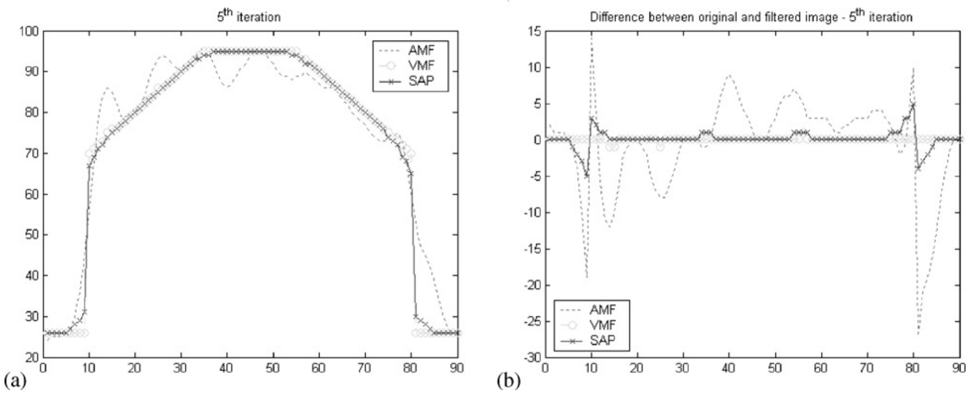


Fig. 10. Line plots of row 45 for the pyramid image corrupted by impulsive noise: (a) outputs of the filters, (b) differences between the original and the filtered image.

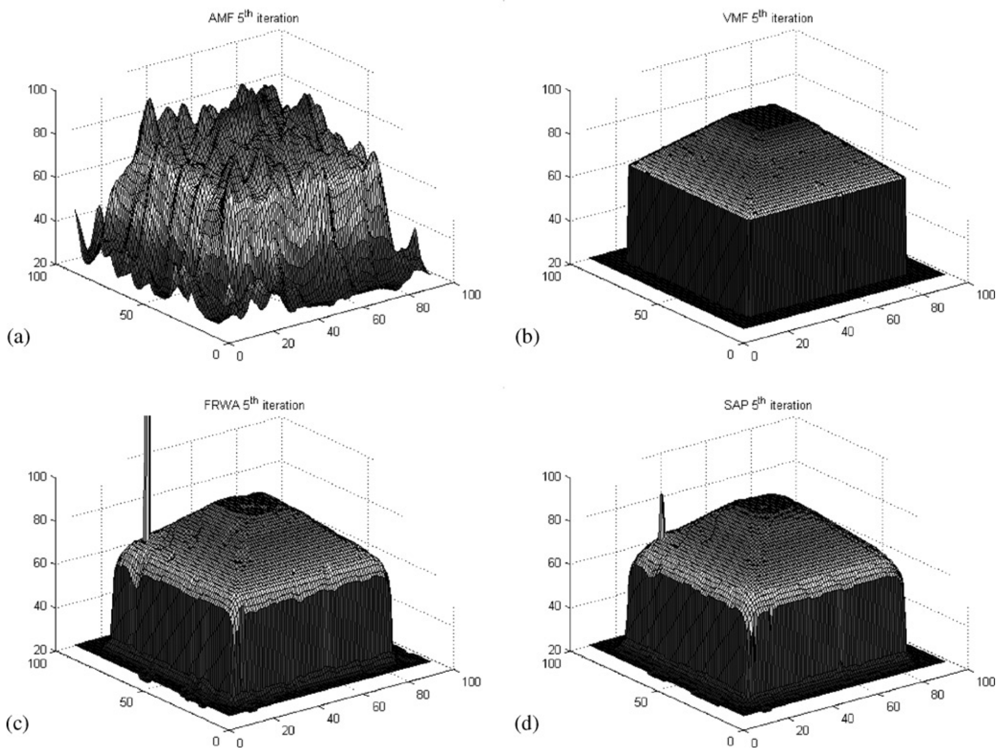


Fig. 11. 3D plots for the pyramid image corrupted by impulsive noise: (a) AMF, (b) VMF, (c) FRWA and (d) SAP, (five iterations).

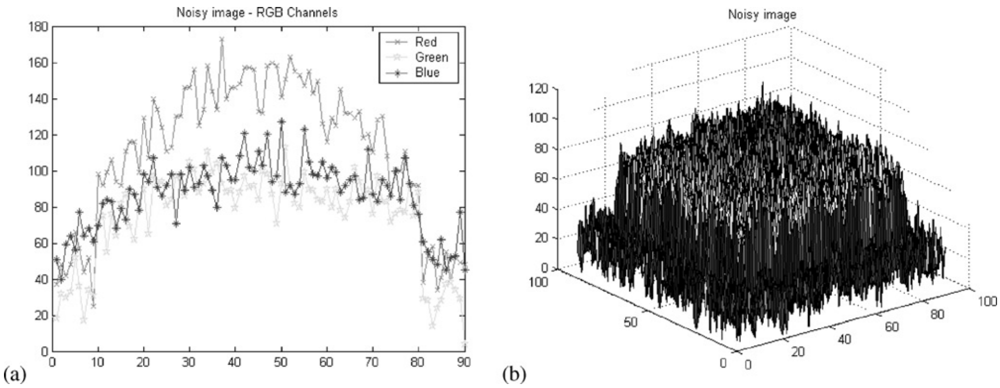


Fig. 12. Pyramid image corrupted by Gaussian noise with $\sigma = 10$: (a) line plot of row 45, (b) 3D plot of green image component.

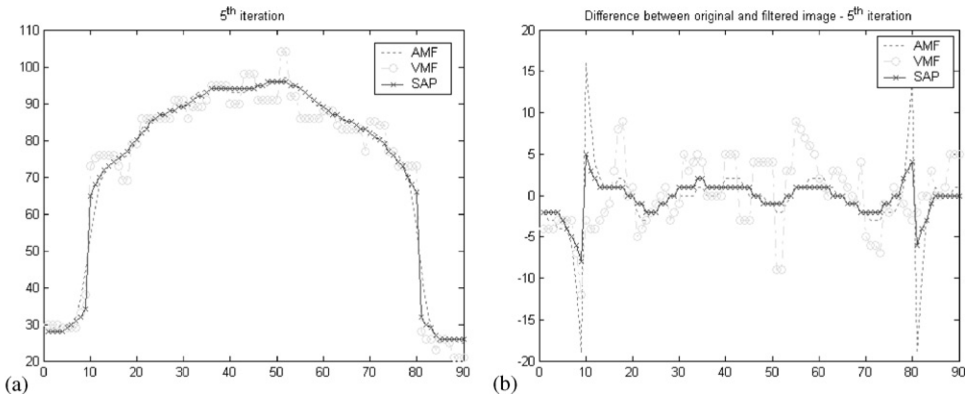


Fig. 13. Line plots of row 45 for the pyramid image corrupted by Gaussian noise: (a) outputs of the filters, (b) differences between the original and the filtered image.

intersection on line 45 for all RGB channels, while Fig. 6b depicts the 3D plot of the green component.

Figs. 7 and 8 depict the edge preservation property of the various filters under consideration when they are applied to the noise-free pyramid image.

In the sequence, the test image was corrupted by multivariate impulsive noise following the model in (31) $p = 0.1$ and $p_1 = p_2 = p_3 = 0.25$. Fig. 9 depicts the corrupted pyramid image, while filtering results are shown in Figs. 10 and 11.

In the next experiment the test examines the filter efficiency in the presence of Gaussian noise near slope edges. The test image was contaminated by additive zero mean Gaussian noise with standard deviation $\sigma = 10$. Fig. 12 shows the corrupted pyramid image, while Figs. 13 and 14 summarize filtering results.

Finally, in a last experiment all algorithms were tested on an image containing a ramp edge corrupted with mixed Gaussian and impulsive noise. The pyramid image was contaminated by additive zero mean Gaussian noise with standard deviation $\sigma = 10$ and then with 10% impulsive noise ($p = 0.1$ and $p_1 = p_2 = p_3 = 0.25$) (Figs. 15–17).

All tested filters, with the exception of the arithmetic mean filter, preserve to a certain degree the uncorrupted image step edge. However, differences in performance could be observed near the corners of the top-cut pyramid. The new filters introduced here are based on weighted averages, and thus the distortion that is introduced at the top-cut corners is less than the one introduced by the VMF. Despite the fact that this seems unlikely, it is easy to understand why it is

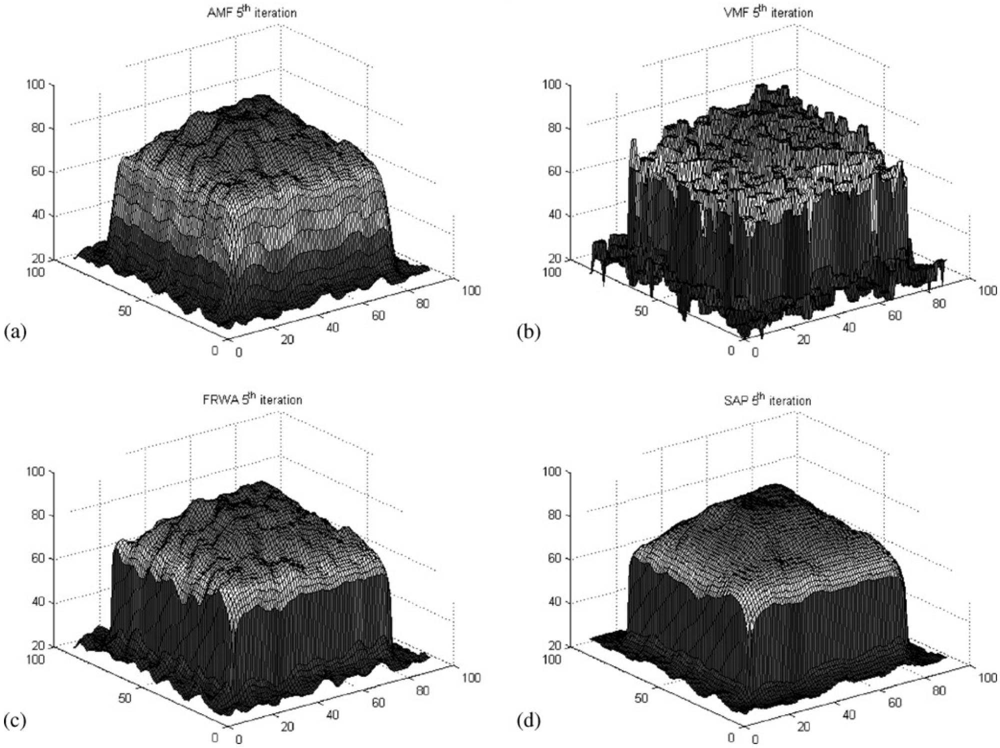


Fig. 14. 3D plots for the pyramid image corrupted by Gaussian noise: (a) AMF, (b) VMF, (c) FRWA and (d) SAP, (five iterations).

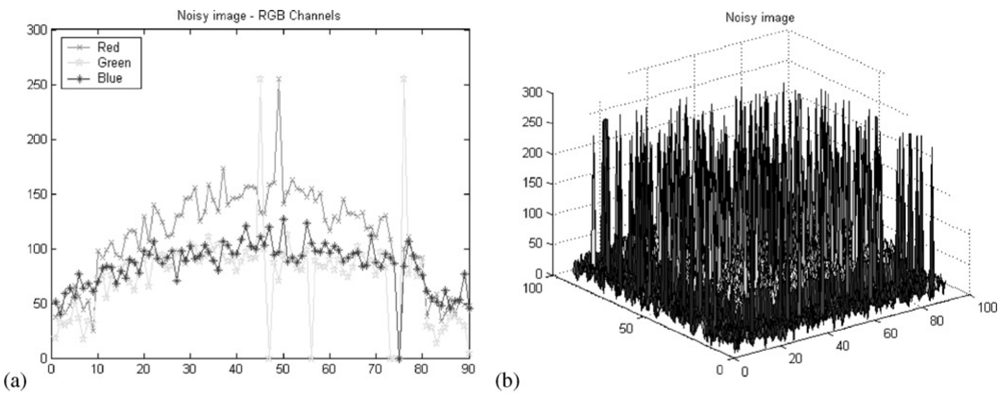


Fig. 15. Pyramid image corrupted by mixed Gaussian noise ($\sigma = 10$) and 10% impulsive noise: (a) line plot of row 45, (b) 3D plot of green image component.

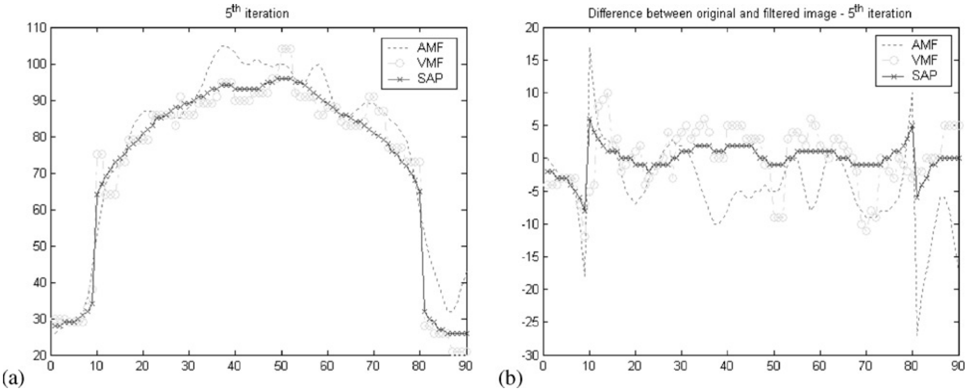


Fig. 16. Line plots of row 45 for the pyramid image corrupted by mixed Gaussian and impulsive noise: (a) outputs of the filters (b) differences between the original and the filtered image.

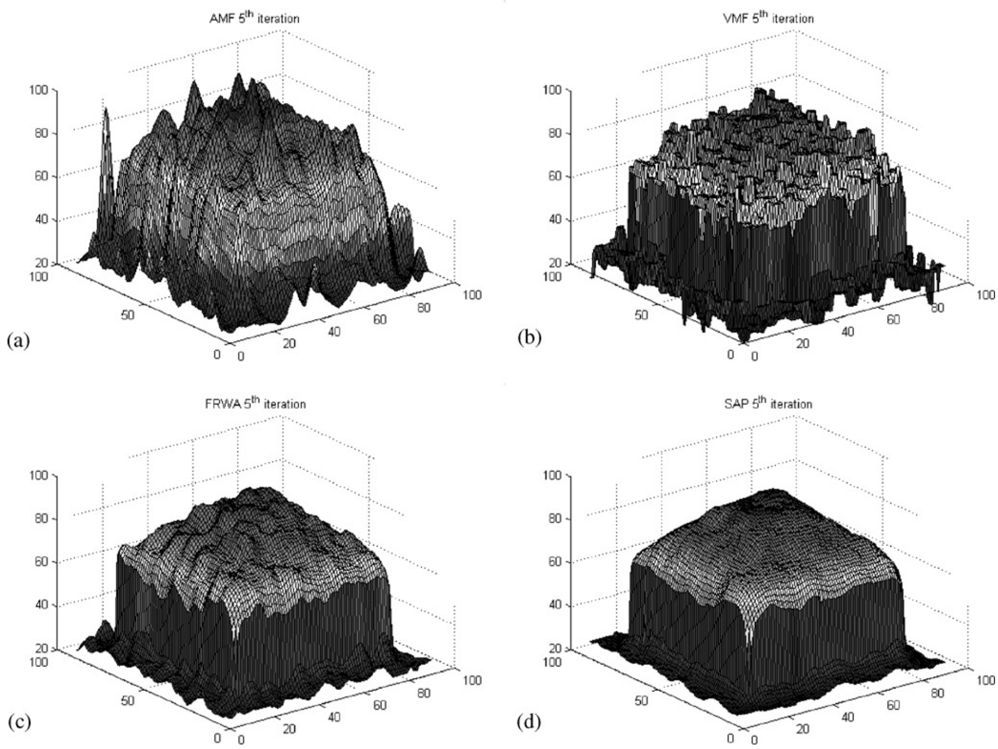


Fig. 17. 3D plots for the pyramid image corrupted by mixed Gaussian and impulsive noise: (a) AMF, (b) VMF, (c) FRWA and (d) SAP, (five iterations).

happening. Pixels on different sides of the edge are weakly connected by any geodesic path, due to the fact that the distance function along a path crossing an edge is very high resulting in a minimal weight. On the other hand, the VMF outputs the most centrally located pixel, and it is possible that at corners it may replace a pixel value with a background sample. In the slope edges in the pyramid image, the geodesic path filters introduce a slight blur (this could be improved by increasing the β value), while the VMF provides the most crispy output leaving much of the structure unchanged (Figs. 7 and 8).

When it comes to edge preservation in noise corrupted images, as it was excepted, the VMF gives best results for images corrupted with impulsive noise only, while results obtained for the AMF are the worst in this case. However, results obtained for the SAP and FRWA filters especially for the top-cut square in the pyramid image are very close to the original (see Figs. 10 and 11).

On the other hand for images corrupted with Gaussian noise, the AMF as expected gave much better results than the VMF, especially in the flat homogeneous regions, but it blurred object edges. Our filtering structure gives superior results in flat regions and near the edges of the image regions (see Figs. 13 and 14).

For images corrupted with mixed Gaussian and impulsive noise neither the VMF nor AMF provide acceptable results. The FRWA and SAP filters performance is excellent. The new filters remove outliers introduced by impulsive noise (weights associated with such pixels are close to zero) and smooth flat regions leaving the edges of the object almost unchanged (see Figs. 16 and 17).

In conclusion, based on the above simulation studies, the following conclusions can be drawn:

(1) The vector median algorithm works little better than the new filter class near slope edges when applied to noise-free images and if only impulsive noise is present.
(2) The VMF fails in the presence of Gaussian noise.
(3) The arithmetic mean filter works well in homogeneous regions with additive Gaussian noise.
(4) The proposed algorithms can suppress Gaussian as well-mixed Gaussian and impulsive noise in homogeneous regions and near edges much better than the AMF and VMF.
(5) The proposed filters induce smoothing effects in the vicinity of edges, stronger near slope edges, and its step edge preservation properties are close to ideal.

Table 3
Noise distributions

Number	Noise model
1	Gaussian ($\sigma = 30$)
2	Impulsive ($p = 0.12$, $p_1 = p_2 = p_3 = 0.3$)
2	Mixed: Gaussian ($\sigma = 30$) + impulsive ($p = 0.12$, $p_1 = p_2 = p_3 = 0.3$)

5.3. Application to natural color images

A number of experiments has been performed in order to evaluate the new filtering framework presented in this paper. The noise attenuation properties of the different filters were examined by utilizing the color test images LENA and PEPPERS (Fig. 23). The test images have been contaminated using various contamination models in order to assess the performance of the filters under different noise scenarios (see Table 3).

The *root mean squared error* (RMSE), the *signal-to-noise ratio* (SNR), the *peak signal-to-noise ratio* (PSNR), the *normalized mean square error* (NMSE) and the *normalized color difference* (NCD) [21] were used for the analysis. All those standard objective quality measures, widely used in color image processing, are defined by the following formulas:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{NML}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \sum_{l=1}^{L} (F^l(i,j) - \hat{F}^l(i,j))^2}, \tag{33}$$

$$\mathrm{NMSE} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \sum_{l=1}^{L} (F^l(i,j) - \hat{F}^l(i,j))^2}{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \sum_{l=1}^{L} F^l(i,j)^2}, \tag{34}$$

$$\mathrm{SNR} = 10 \log \left[\frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \sum_{l=1}^{L} F^l(i,j)^2}{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \sum_{l=1}^{L} (F^l(i,j) - \hat{F}^l(i,j))^2} \right], \tag{35}$$

Table 4
Filters taken for comparison with the proposed noise reduction technique

Notation	Method	Ref.
AMF	Arithmetic mean filter	[21]
VMF	Vector median filter	[1]
BVDF	Basic vector directional filter	[30]
GVDF	Generalized vector directional filter	[29]
DDF	Directional-distance filter	[7]
HDF	Hybrid directional filter	[6]
AHDF	Adaptive hybrid directional filter	[6]
FVDF	Fuzzy vector directional filter	[17]
ANNF	Adaptive nearest-neighbor filter	[16]
ANP-EF	Adaptive nonparametric (Exponential) filter	[19]
ANP-GF	Adaptive nonparametric (Gaussian) filter	[19]
ANP-DF	Adaptive nonparametric (Directional) filter	[19]
VBAMMF	Vector Bayesian adaptive median/mean filter	[19]

$$\mathrm{PSNR} = 20 \log \left(\frac{255}{\mathrm{RMSE}} \right), \tag{36}$$

where M, N are the image dimensions, and $F^l(i,j)$ and $\hat{F}^l(i,j)$ denote the lth component of the original image vector and its estimation at pixel (i,j), respectively.

The NCD perceptual measure is evaluated over the uniform $L^*u^*v^*$ color space. This difference measure is defined as follows:

$$\mathrm{NCD} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Delta E_{Luv}}{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} E_{Luv}^*}, \tag{37}$$

where $\Delta E_{Luv} = [(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2]^{1/2}$ is the perceptual color difference and $E_{Luv}^* = [(L^*)^2 + (u^*)^2 + (v^*)^2]^{1/2}$ is the *norm* or *magnitude* of the uncorrupted original image pixel in the $L^*u^*v^*$ space.

The performance of new filters is compared with the performance of a number of standard color image processing filters listed in Table 4.

Results obtained using the new filtering techniques in comparison with the standard filtering algorithms are collected in Tables 5–10. Additionally, Figs. 24 and 25 show the comparison of the new filtering technique with the standard vector median.

In the tables, following notation is used: SAP-2, 3 denote the SAP filter with 2 and 3 steps, respectively, SAP-AD denotes the adaptive version of SAP-2 ($n=2$) while FRWA-2, 3 denote the FRWA filter, FRWA-AD denotes adaptive version of FRWA-2. Subscripts denote the number of performed iterations. For all reference filters the best iteration in terms of PSNR is presented. In all test images predefined parameter values were used. Namely, the parameter values $\beta = 13$, $\alpha = 1.2$ were used in the SAP-2, SAP-3 and in the FRWA. The *SAP-AD* and *FRWA-AD* filters use $\gamma = 6$.

Additionally in Tables 6 and 9 presenting results of applying tested filters to images corrupted with impulsive noise SAP-2* and FRWA-2* denote appropriate filters when optimal values of parameters were used.

In our experiments a wide range of filter parameters was evaluated. Figs. 18 and 19 depict the efficiency of the proposed algorithms in terms of PSNR and NCD quality measures, as a function of the design parameters α and β. It can be easily noticed that both algorithms yield comparable results with a flat maximum of PSNR (minimum of NCD), which ensures their robustness to optimal parameter settings. The comparison of the new filter efficiency with some standard noise suppression techniques is presented in Fig. 20, where the PSNR and NCD dependency on the amount of mixed impulsive and Gaussian noise is shown. For all compared filters the best restoration result obtained in a series of iterations was chosen.

Figs. 21 and 22 show the PSNR, SNR, NMSE and NCD dependence on the design parameter γ for adaptive versions of our filter (SAP-AD and FRWA-AD), when applied to LENNA and PEPPERS test images with mixed Gaussian and impulsive noise, respectively (Figs. 24b and 25b).

Tables 5–10 indicate that the new filters outperform existing filters for the Gaussian as well as Gaussian and impulsive noise. The best results for Gaussian noise attenuation for the majority of existing filters are obtained after many iterations, while for filters based on the geodesic paths' concept the best results are obtained after two or three iterations. The new filtering techniques give superior results in terms of objective quality measures and in terms of visual appearance. In addition to excellent noise attenuation properties, our filters produce images with well-preserved and even enhanced edges, while filters such as the VMF produce visible color clusters (Figs. 24 and 25).

The effectiveness of the new filters is particularly clear in the case of mixed noise corruption (see Tables 7 and 10 and Figs. 26 and 27), where the new filters are

Table 5
Comparison of the new algorithms with the standard techniques (Table 4) using the LENA standard image corrupted by Gaussian noise $\sigma = 30$ (subscript denotes the iteration number)

Method_N	NMSE (10^{-3})	RMSE	SNR (dB)	PSNR (dB)	NCD (10^{-4})
None	420.550	29.075	13.762	18.860	250.090
AMF_1	66.452	11.558	21.775	26.873	95.347
VMF_5	87.314	13.248	20.589	25.688	117.170
BVDF_3	279.540	23.705	15.536	20.634	117.400
GVDF_5	76.713	12.418	21.151	26.250	84.876
DDF_5	100.500	14.213	19.979	25.077	108.960
HDF_5	66.584	11.569	21.766	26.865	92.769
AHDF_5	60.166	10.997	22.206	27.305	91.369
FVDF_3	57.466	10.748	22.406	27.504	77.111
ANNF_3	63.341	11.284	21.983	27.082	82.587
ANP-E_3	60.396	11.018	22.190	27.288	76.896
ANP-G_3	60.443	11.023	22.187	27.285	76.890
ANP-D_3	58.389	10.834	22.337	27.435	78.486
SAP-2_1	51.876	10.212	22.850	27.949	80.838
SAP-2_2	45.043	9.515	23.464	28.562	69.260
SAP-2_3	47.880	9.810	23.198	28.297	68.355
SAP-3_1	73.903	12.188	21.313	26.412	100.080
SAP-3_2	51.015	10.127	22.923	28.022	78.633
SAP-3_3	48.027	9.826	23.185	28.284	73.222
SAP-AD_1	52.093	10.233	22.832	27.931	81.007
SAP-AD_2	45.540	9.568	23.416	28.515	68.912
SAP-AD_3	49.343	9.959	23.068	28.166	68.150
FRWA-2_1	87.030	13.227	20.603	25.702	110.020
FRWA-2_2	50.823	10.107	22.939	28.038	79.661
FRWA-2_3	46.632	9.682	23.313	28.412	73.902
FRWA-3_1	90.815	13.511	20.418	25.517	108.910
FRWA-3_2	50.807	10.106	22.941	28.039	78.547
FRWA-3_3	47.760	9.798	23.209	28.308	73.568
FRWA-AD_1	85.854	13.137	20.662	25.761	109.950
FRWA-AD_2	50.405	10.066	22.975	28.074	79.322
FRWA-AD_3	46.857	9.705	23.292	28.391	73.594

compared with the VMF and DDF using three color test images: LENA, PEPPERS and GOLDHILL.

It should be pointed out that in case of images slightly corrupted by Gaussian or mixed Gaussian and impulsive noise, the AMF obtains the best quantitative results. However, visual inspection reveals that results obtained via our methods look subjectively better. As the intensity of noise increases, the quantitative results obtained through our filters become significantly better than those obtained by any other filter (see Fig. 20).

For impulsive noise as expected the VMF utilizing L_2 norm gives best results, although the FRWA filter utilizing a path of length $n = 2$ is in some cases as good as VMF. This is due to the fuzzy construction of the new filter class, which is not well suited for pure, low intensity impulsive noise removal, due to the lack of incorporated noise detecting module.

Table 6
Comparison of the new algorithms with the standard techniques (Table 4) using the LENA standard image corrupted by impulsive noise ($p = 0.12$, $p_1 = p_2 = p_3 = 0.3$) (subscript denotes the iteration number)

Method_N	NMSE (10^{-3})	RMSE	SNR (dB)	PSNR (dB)	NCD (10^{-4})
None	474.400	30.881	13.239	18.337	100.480
AMF_1	75.815	12.345	21.202	26.301	101.740
$\text{VMF}(L_1)_1$	16.303	5.725	27.877	32.976	39.771
$\text{VMF}(L_2)_1$	17.663	5.959	27.529	32.628	40.252
BVDF_1	22.807	6.771	26.419	31.58	41.333
GVDF_1	19.474	6.257	27.105	32.204	41.773
DDF_1	18.318	6.068	27.371	32.470	40.186
HDF_1	18.610	6.116	27.303	32.401	41.275
AHDF_1	18.310	6.067	27.373	32.472	41.166
FVDF_1	22.251	6.688	26.527	31.625	44.686
ANNF_1	26.800	7.340	25.719	30.817	48.009
ANP-E_1	78.601	12.570	21.046	26.144	82.457
ANP-G_1	78.623	12.571	21.045	26.143	82.478
ANP-D_1	24.178	6.971	26.166	31.264	46.070
SAP-2_1	24.421	7.006	26.122	31.221	50.512
SAP-2_2	30.173	7.788	25.204	30.302	51.412
SAP-2_1^*	24.169	6.970	26.167	31.266	49.606
SAP-2_2^*	29.194	7.661	25.347	30.446	50.472
SAP-AD_1	23.936	6.936	26.210	31.308	49.967
SAP-AD_1	29.324	7.678	25.328	30.426	50.686
FRWA-2_1	22.630	6.745	26.453	31.552	52.290
FRWA-2_2	21.375	6.555	26.701	31.800	45.748
FRWA-2_1^*	26.237	7.262	25.811	30.909	49.178
FRWA-2_2^*	20.055	6.349	26.978	32.076	42.940
FRWA-AD_1	22.446	6.717	26.489	31.587	51.497
FRWA-AD_2	21.005	6.498	26.777	31.875	44.986

Filters based on a larger supporting element W, such as the SAP filter, yield worse performance for impulsive noise but give superior results for images contaminated with Gaussian noise. This is a well-known property of ranked-order filters which also applies to our filtering schemes. In general larger in size supporting window W, and thus a larger geodesic path, will result in extensive smoothing. However, unlike fixed window, ranked-type filters, the proposed here filters control the degree of smoothing by changing appropriately the parameter β during successive iterations. For example, a high β value in the first iteration, results in aggressive outlier removal at the expense of lower performance in terms of quality measures such as PSNR or NCD. However, using a cooling value $\alpha < 1$, which leads to lower β values in subsequent iterations, visually sharpened images and excellent quantitative results are obtained. Similar behavior can be obtained by tuning the parameter value γ when the adaptive version of the algorithm is considered (see Tables 6 and 9).

In conclusion, from the results listed in the tables, it can be easily seen that the new filters, especially the FRWA filter, provide consistently good results in all types of the noise, outperforming the other multichannel filters under consideration. The FRWA filter

Table 7
Comparison of new algorithms with standard techniques using LENA image corrupted by 12% impulse and Gaussian noise $\sigma = 30$ (subscript denotes the iteration number)

Method_N	NMSE (10^{-3})	RMSE	SNR (dB)	PSNR (dB)	NCD (10^{-4})
None	905.930	42.674	10.429	15.528	305.550
AMF_3	97.444	13.996	20.112	25.211	95.800
VMF_5	96.464	13.925	20.156	25.255	121.790
BVDF_3	336.460	26.006	14.731	19.829	123.930
GVDF_5	91.118	13.534	20.404	25.503	89.277
DDF_5	110.620	14.912	19.561	24.660	113.390
HDF_5	74.487	12.236	21.279	26.378	97.596
AHDF_5	68.563	11.740	21.639	26.738	96.327
FVDF_3	73.796	12.179	21.320	26.418	83.629
ANNF_3	75.652	12.332	21.212	26.310	86.836
ANP-E_3	90.509	13.488	20.433	25.532	97.621
ANP-G_3	90.523	13.489	20.432	25.531	97.603
ANP-D_3	74.203	12.213	21.296	26.394	85.026
SAP-2_1	64.287	11.368	21.919	27.017	86.841
SAP-2_2	50.048	10.030	23.006	28.105	72.783
SAP-2_3	51.180	10.143	22.909	28.008	71.245
SAP-3_1	58.478	10.842	22.330	27.429	79.085
SAP-3_2	54.580	10.474	22.630	27.728	71.575
SAP-3_3	58.917	10.883	22.298	27.396	71.859
SAP-AD_1	64.388	11.377	21.912	27.010	87.087
SAP-AD_2	50.575	10.083	22.961	28.059	72.615
SAP-AD_3	52.611	10.284	22.789	27.888	71.201
FRWA-2_1	111.760	14.988	19.517	24.616	118.950
FRWA-2_2	60.190	11.000	22.205	27.303	85.175
FRWA-2_3	53.167	10.338	22.744	27.842	78.516
FRWA-3_1	123.260	15.741	19.092	24.190	119.910
FRWA-3_2	60.665	11.042	22.171	27.270	84.690
FRWA-3_3	54.572	10.474	22.630	27.729	78.638
FRWA-AD_1	116.040	15.272	19.354	24.453	121.780
FRWA-AD_2	63.225	11.273	21.991	27.090	86.255
FRWA-AD_3	54.950	10.510	22.600	27.699	78.643

could be used as universal filter able to attenuate different types of noise with image detail and edge preservation.

6. Conclusions

In this paper a new class of algorithms for filtering color image data has been introduced. These filters utilize fuzzy membership functions over vectorial inputs connected via geodesic paths to adapt to local image features. The behavior of the new filters was analyzed and their performance was compared with the commonly used color image filters.

Experiments on the synthetic and natural images reveal that the new filters induce slight smoothing effect in the vicinity of edges, but its strength

Table 8
Comparison of the new algorithm with the standard techniques using the PEPPERS standard image corrupted by Gaussian noise $\sigma = 30$ (subscript denotes the iteration number)

Method_N	NMSE (10^{-3})	RMSE	SNR (dB)	PSNR (dB)	NCD (10^{-4})
None	502.410	28.683	12.989	18.978	244.190
AMF_3	88.815	12.060	20.515	26.504	99.043
VMF_5	105.180	13.124	19.781	25.770	123.390
BVDF_5	367.740	24.539	14.345	20.334	124.350
GVDF_3	99.400	12.758	20.026	26.015	97.348
DDF_5	118.820	13.949	19.251	25.240	114.400
HDF_5	79.698	11.424	20.986	26.975	101.140
AHDF_5	72.331	10.883	21.407	27.396	99.673
FVDF_3	72.888	10.925	21.373	27.362	89.743
ANNF_3	80.934	11.512	20.919	26.908	96.789
ANP-E_3	79.688	11.423	20.986	26.975	100.860
ANP-G_3	79.674	11.422	20.987	26.976	100.850
ANP-D_3	73.211	10.949	21.354	27.343	89.078
SAP-2_1	65.991	10.395	21.805	27.794	101.000
SAP-2_2	54.659	9.461	22.623	28.612	91.343
SAP-2_3	56.886	9.651	22.450	28.439	91.386
SAP-3_1	61.486	10.034	22.112	28.101	95.622
SAP-3_2	59.335	9.857	22.267	28.256	91.650
SAP-3_3	65.249	10.337	21.854	27.843	93.237
SAP-AD_1	64.935	10.312	21.875	27.864	98.429
SAP-AD_2	55.061	9.495	22.592	28.581	89.165
SAP-AD_3	58.824	9.814	22.304	28.293	89.858
FRWA-2_1	113.400	13.627	19.454	25.443	126.950
FRWA-2_2	63.888	10.228	21.946	27.935	99.028
FRWA-2_3	58.818	9.814	22.305	28.294	95.479
FRWA-3_1	118.160	13.910	19.275	25.264	126.590
FRWA-3_2	63.910	10.230	21.944	27.933	98.983
FRWA-3_3	60.045	9.916	22.215	28.204	96.033
FRWA-AD_1	112.240	13.557	19.498	25.487	125.070
FRWA-AD_2	63.121	10.167	21.998	27.987	96.364
FRWA-AD_3	58.500	9.787	22.328	28.317	92.844

depends on the type of edge: stronger smoothing effect can be observed near slope edges and very weak near step edges and this process is not significantly amplified when iterating, moreover, step edges are going to be sharpened during the iteration process

The new filters, while iterating, yield piecewise constant solutions, but it does not induce edge dislocation effects, as the proposed method can be seen as a special case of anisotropic diffusion, which has the nice property of not dislocating edges, in contrast to the scale-space approaches.

Table 9
Comparison of the new algorithms with the standard techniques (Table 4) using the PEPPERS standard image corrupted by impulsive noise ($p = 0.12$, $p_1 = p_2 = p_3 = 0.3$) (subscript denotes the iteration number)

Method_N	NMSE (10^{-3})	RMSE	SNR (dB)	PSNR (dB)	NCD (10^{-4})
None	618.340	31.820	12.088	18.077	93.777
AMF_1	105.490	13.143	19.768	25.757	116.390
$\text{VMF}(L_1)_1$	25.401	6.449	25.951	31.940	51.152
$\text{VMF}(L_2)_1$	26.422	6.578	25.780	31.769	51.465
BVDF_1	57.583	9.710	22.397	28.386	55.796
GVDF_1	31.856	7.222	24.968	30.957	52.701
DDF_1	29.684	6.972	25.275	31.264	51.547
HDF_1	26.819	6.627	25.716	31.705	51.424
AHDF_1	26.430	6.579	25.779	31.768	51.317
FVDF_1	33.337	7.388	24.771	30.760	54.074
ANNF_1	45.115	8.595	23.457	29.446	65.891
ANP-DF_1	37.240	7.809	24.290	30.279	56.393
ANP-EF_1	106.700	13.218	19.718	25.707	99.762
ANP-GF_1	106.690	13.218	19.719	25.708	99.745
SAP-2_1	31.685	7.203	24.992	30.981	59.948
SAP-2_2	34.556	7.522	24.615	30.604	59.976
SAP-AD_1	32.286	7.271	24.910	30.899	58.700
SAP-AD_2	35.214	7.594	24.533	30.522	58.300
FRWA-2_1	34.337	7.498	24.642	30.631	64.126
FRWA-2_2	26.387	6.573	25.786	31.775	53.454
FRWA-2_1^*	34.753	7.544	24.590	30.579	61.614
FRWA-2_2^*	25.105	6.412	26.002	31.991	51.596
FRWA-AD_1	35.002	7.571	24.559	30.548	63.902
FRWA-AD_2	27.267	6.682	25.644	31.633	52.830

The new filter class based on digital paths and connection cost can be seen as a powerful generalization of the multichannel anisotropic diffusion presented [3,13] and an extension of the fuzzy adaptive filters [18,19].

The path connection costs evaluated over all possible digital paths, are used to derive fuzzy membership functions that quantify the similarity between vectorial inputs. The proposed filtering structure is then using the function outputs to appropriately weight input contributions in order to determine the filtering result. The proposed filtering schemes parallelize the familiar structure of the adaptive multichannel filter introduced in [18] and they can successfully eliminate Gaussian, impulsive as well as mixed-type noise.

Simulation results indicate that the proposed filters' performance expressed through standard image restoration measures is superior to the examined filters for the noise models under consideration. Moreover, the analysis of the computational complexity shows that some of the presented filtering techniques are faster than the optimized version of the vector median filter.

Table 10
Comparison of new algorithms with standard techniques using PEPPERS image corrupted by 12% impulse and Gaussian noise $\sigma = 30$ (subscript denotes the iteration number)

Method_N	NMSE (10^{-3})	RMSE	SNR (dB)	PSNR (dB)	NCD (10^{-4})
None	1052.000	41.505	9.780	15.769	191.590
AMF_3	125.890	14.357	19.000	24.989	118.480
VMF_5	84.436	11.758	20.735	26.724	85.847
BVDF_5	193.870	17.817	17.125	23.114	88.756
GVDF_5	113.300	13.621	19.458	25.447	82.889
DDF_5	88.961	12.069	20.508	26.497	82.072
HDF_5	77.452	11.262	21.110	27.099	81.870
AHDF_5	73.811	10.994	21.319	27.308	81.235
FVDF_3	87.091	11.942	20.600	26.589	86.830
ANNF_3	96.695	12.583	20.146	26.135	91.361
ANP-E_3	120.470	14.045	19.191	25.180	120.610
ANP-G_3	120.450	14.044	19.192	25.181	120.590
ANP-D_3	88.051	12.008	20.553	26.542	88.486
SAP-2_1	90.496	12.173	20.434	26.423	94.373
SAP-2_2	65.532	10.359	21.835	27.824	88.880
SAP-2_3	64.212	10.254	21.924	27.913	90.492
SAP-3_1	75.521	11.120	21.219	27.208	91.964
SAP-3_2	66.369	10.425	21.780	27.769	90.512
SAP-3_3	70.382	10.735	21.525	27.514	92.893
SAP-AD_1	85.337	11.821	20.689	26.678	92.370
SAP-AD_2	64.778	10.299	21.886	27.875	87.451
SAP-AD_3	65.231	10.335	21.855	27.845	89.520
FRWA-2_1	151.560	15.754	18.194	24.183	107.190
FRWA-2_2	84.341	11.752	20.740	26.729	89.887
FRWA-2_3	73.304	10.956	21.349	27.338	90.240
FRWA-3_1	180.200	17.178	17.442	23.431	112.410
FRWA-3_2	85.053	11.801	20.703	26.692	92.542
FRWA-3_3	73.344	10.959	21.346	27.335	92.549
FRWA-AD_1	149.800	15.662	18.245	24.234	106.180
FRWA-AD_2	82.738	11.640	20.823	26.812	87.809
FRWA-AD_3	72.223	10.875	21.413	27.402	88.034

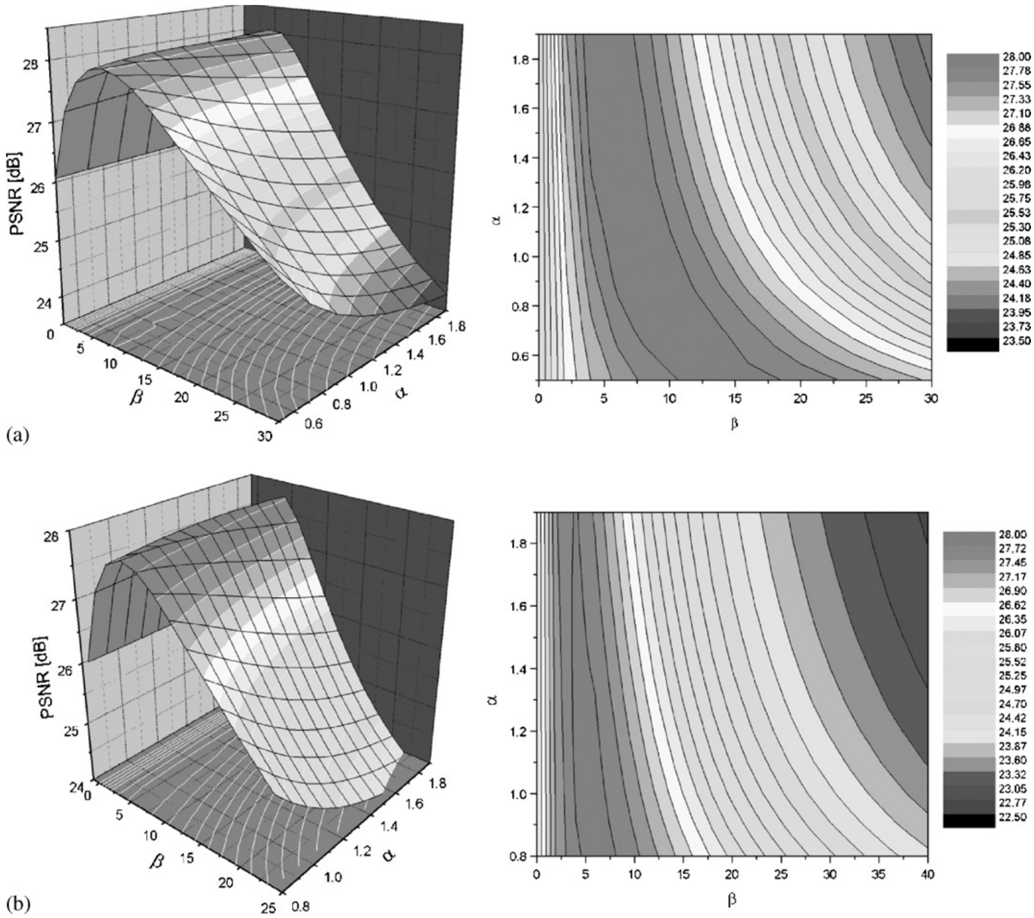


Fig. 18. Efficiency of the (a) SAP and (b) FRWA in terms of PSNR and its dependence on the α and β values for LENA standard image corrupted by 12% impulsive and Gaussian noise, ($\sigma = 30$) ($n = 2$, 3 iterations).

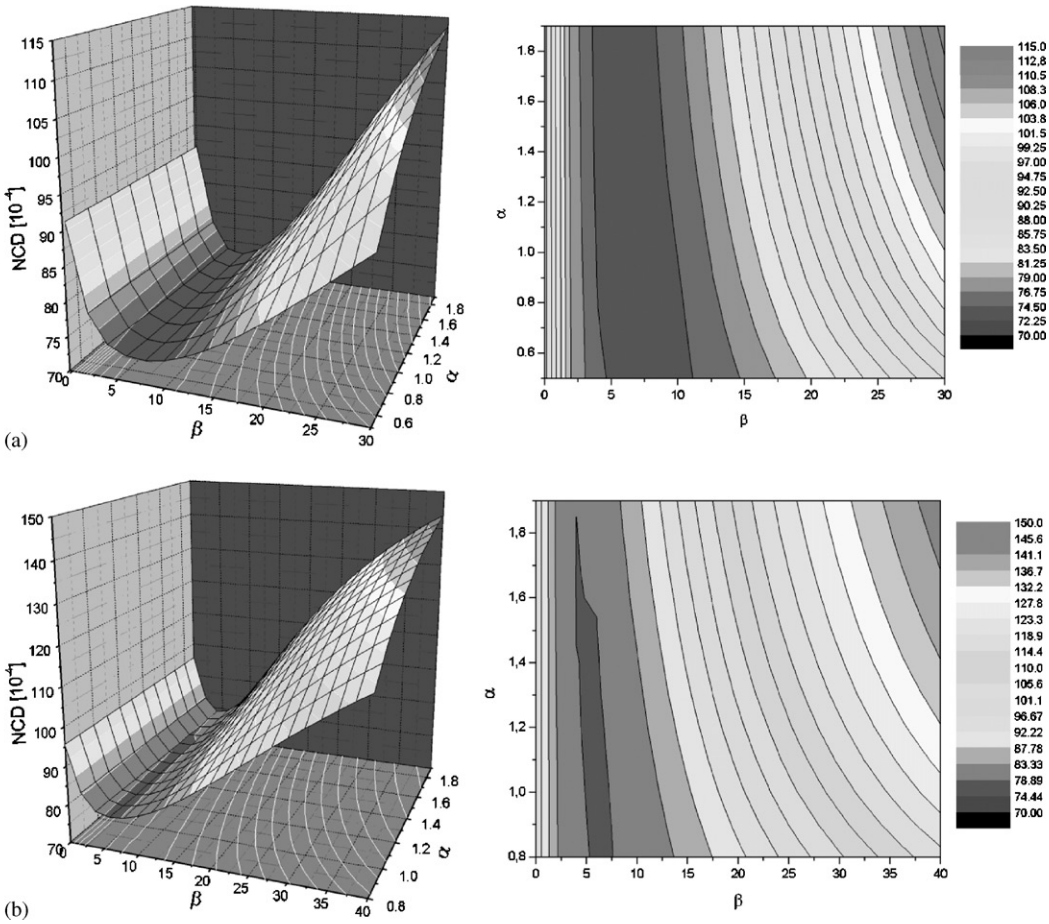
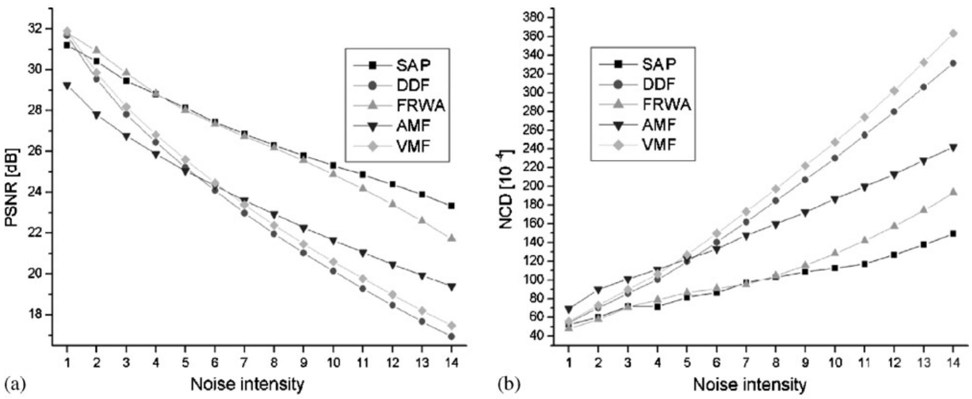


Fig. 19. Efficiency of the (a) SAP and (b) FRWA in terms of NCD and its dependence on the α and β values for LENA standard image corrupted by 12% impulsive and Gaussian noise ($\sigma = 30$) ($n = 2$, 3 iterations).



Noise intensity	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gaussian σ	5	10	15	20	25	30	35	40	45	50	55	60	65	70
Impulsive [%]	1	2	3	4	5	6	7	8	9	10	11	12	13	14

(c)

Fig. 20. Comparison of standard filters efficiency in terms of the (a) PSNR and the (b) NCD with the new ones for different amounts of noise (mixed Gaussian and impulsive noise—intensities are shown in (c)).

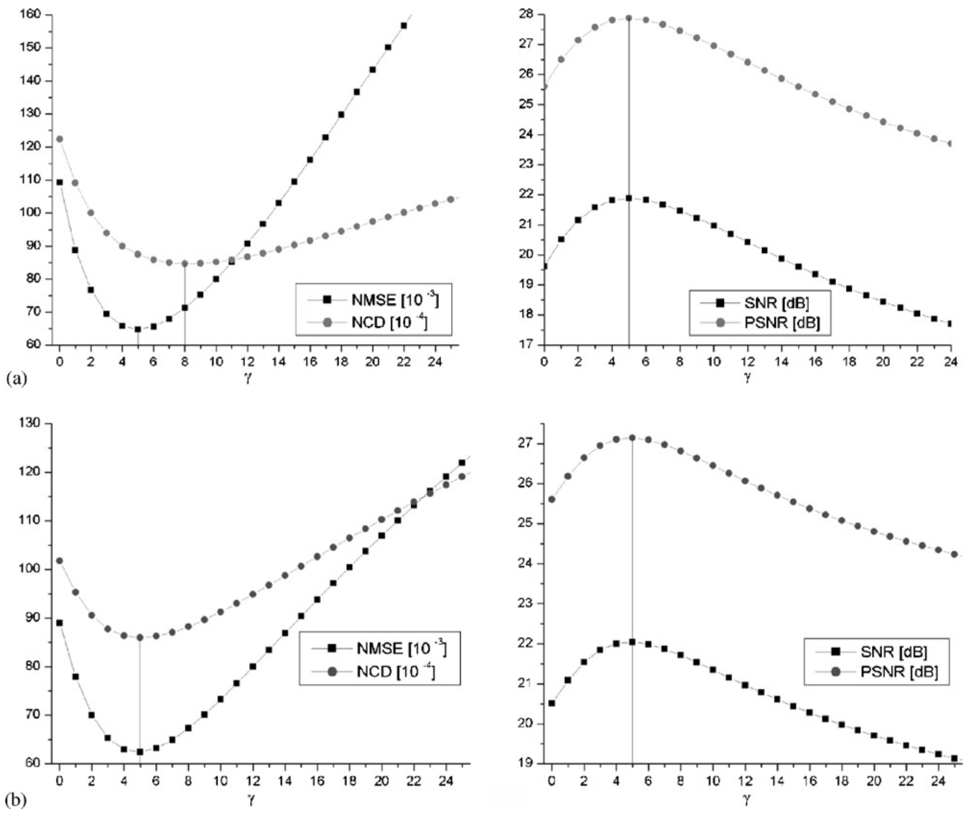


Fig. 21. Efficiency of the (a) adaptive SAP and (b) adaptive FRWA in terms of PSNR, SNR, NCD and NMSE for LENA standard image corrupted by 12% impulsive and Gaussian noise, ($\sigma = 30$) ($n = 2$, 2 iterations).

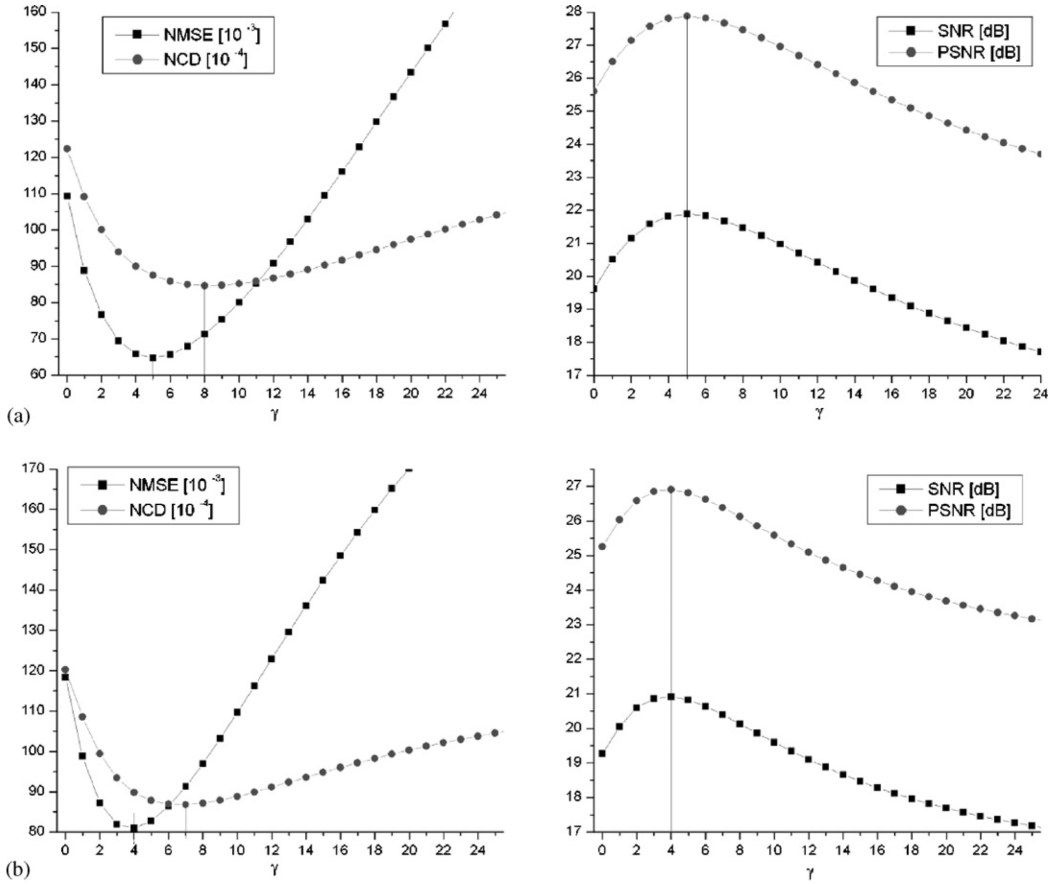


Fig. 22. Efficiency of the (a) adaptive SAP and (b) adaptive FRWA in terms of PSNR, SNR, NCD and NMSE for PEPPERS standard image corrupted by 12% impulsive and Gaussian noise ($\sigma = 30$) ($n = 2$, 2 iterations).

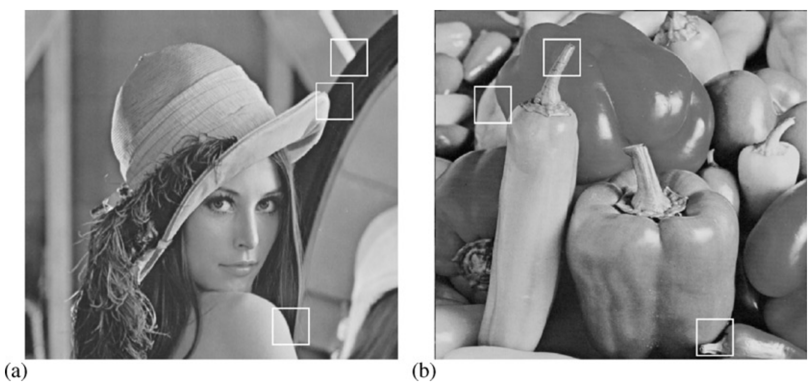


Fig. 23. The test color images (a) LENA and (b) PEPPERS.

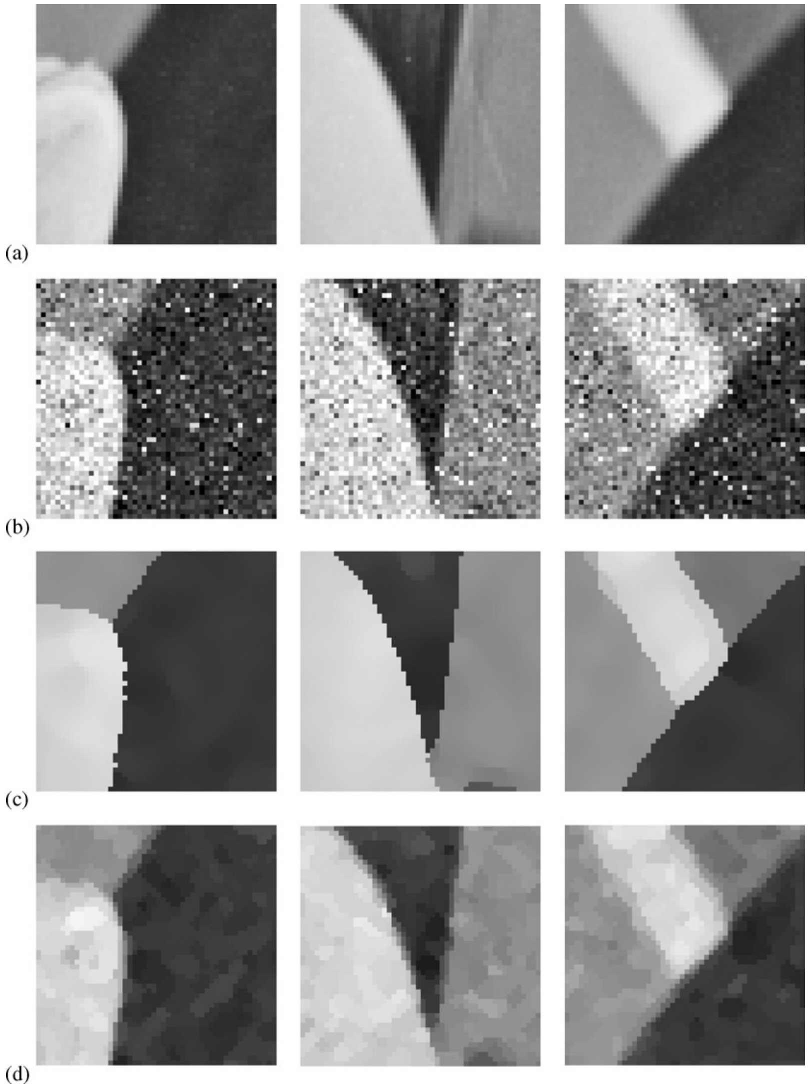


Fig. 24. Comparison of the efficiency of the new filters with the VMF: (a) parts of LENA original image, (b) images corrupted with mixed Gaussian ($\sigma = 30$) and impulsive noise ($p = 0.12$, $p_1 = p_2 = p_3 = 0.3$), (c) images restored with FRWA ($\beta = 10$, $\alpha = 1.25$, 5 iterations) (d) images restored with VMF (3×3 mask, 5 iterations).

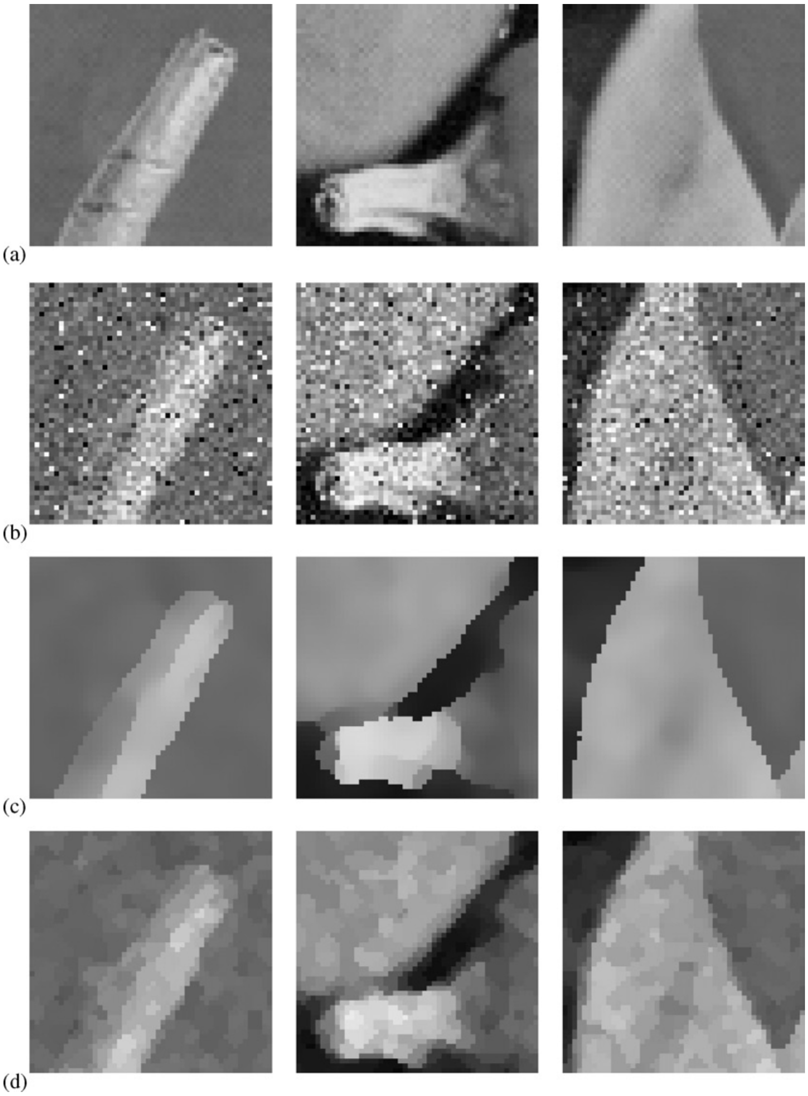


Fig. 25. Comparison of the efficiency of the new filters with the VMF: (a) parts of PEPPERS original image, (b) images corrupted with mixed Gaussian ($\sigma = 30$) and impulsive noise ($p = 0.12$, $p_1 = p_2 = p_3 = 0.3$), (c) images restored with FRWA ($\beta = 10$, $\alpha = 1.25$, 5 iterations) (d) images restored with VMF (3×3 mask, 5 iterations).

Fig. 26. Comparison of the efficiency of the new filters with the standard ones: (a) test images (LENA, PEPPERS and GOLDHILL), (b) images corrupted with mixed Gaussian ($\sigma = 30$) and impulsive noise (5%), (c) result of the standard vector median filtering (3×3 mask), (d) result of the DDF filtering (3×3 mask), (e) result of the FRWA filtering ($\beta = 10, \alpha = 1.25, n = 2$, 5 iterations).

Fig. 27. Comparison of the efficiency of the new filters with the standard ones: (a) test images (LENA, PEPPERS and GOLDHILL), (b) images corrupted with mixed Gaussian ($\sigma = 60$) and impulsive noise (10%), (c) result of the standard vector median filtering (3×3 mask), (d) result of the DDF filtering (3×3 mask), (e) result of the FRWA filtering ($\beta = 10, \alpha = 1.25, n = 2$, 5 iterations).

References

[1] J. Astola, P. Haavisto, Y. Neuovo, Vector median filters, in: IEEE Proceedings, Vol. 78, 1990, pp. 678–689.

[2] K. Barni, V. Cappellini, On the computational complexity of multivariate median filters, Signal Processing 71 (1) (1998) 45–54.

[3] M.J. Black, G. Sapiro, D.H. Marimont, D. Heeger, Robust anisotropic diffusion, IEEE Trans. Image Process. 7 (3) (1998) 421–432.

[4] G. Borgefors, Distance transformations in digital images, Comput. Vision Graph. Image Process. 34 (1986) 344–371.

[5] O. Cuisenaire, Distance transformations: fast algorithms and applications to medical image processing, Ph.D. Thesis, Universite Catholique de Louvain, October 1999.

[6] M. Gabbouj, F.A. Cheickh, Vector median—vector directional hybrid filter for colour image restoration, in: Proceedings of EUSIPCO, 1996, pp. 879–881.

[7] D. Karakos, P.E. Trahanias, Generalized multichannel image filtering structures, IEEE Trans. Image Process. 6 (7) (1997) 1038–1045.

[8] R. Lukac, Optimised directional distance filter, Mach. Graph. Visions: Special Issue on Colour Image Process. Appl., 17 (2/3) (2003) 311–326.

[9] R. Lukac, Introducing of the weight concept to vector directional filters, J. Electrical Eng. 52 (3–4) (2001) 98–100.

[10] R. Lukac, Adaptive impulse noise filtering by using center-weighted directional information, in: Proceedings of the CGIV 2002, France, 2002, pp. 86–89.

[11] R. Lukac, Color image filtering by vector directional order-statistics, Pattern Recognition Image Anal. 12 (3) (2002) 279–285.

[12] N. Madras, G. Slade, The Self-Avoiding Walk, Birkhauser, Boston, 1993.

[13] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 629–639.

[14] I. Pitas, P. Tsakalides, Multivariate ordering in color image processing, IEEE Trans. Circuits Systems Video Technol. 1 (3) (1991) 247–256.

[15] I. Pitas, A.N. Venetsanopoulos, Nonlinear Digital Filters: Principles and Applications, Kluwer Academic Publishers, Boston, MA, 1990.

[16] K.N. Plataniotis, D. Androutsos, V. Sri, A.N. Venetsanopoulos, A nearest neighbor multichannel filter, Electron. Lett. 31 (22) (1995) 1910–1911.

[17] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Fuzzy adaptive filters for multichannel image processing, Signal Process. J. 55 (1) (1996) 93–106.

[18] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Adaptive fuzzy systems for multichannel signal processing, Proc. IEEE 87 (9) (1999) 1601–1622.

[19] K.N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, A.N. Venetsanopoulos, Color image processing using adaptive multichannel filters, IEEE Trans. Image Process. 6 (7) (1997) 933–950.

[20] K.N. Plataniotis, A.N. Venetsanopoulos, Vector filtering, in: S.J. Sangwine, R.E.N. Horne (Eds.), The Colour Image Processing Handbook, Chapman & Hall, Cambridge, Great Britain, 1998, pp. 188–209.

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

[22] M. Schmitt, Mathematical Morphology, Cambridge University Press, Cambridge, May 2001.

[23] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, New York, November 1997.

[24] B. Smolka, A. Chydzinski, K. Plataniotis, A.N. Venetsanopoulos, New filtering technique for the impulsive noise reduction in color images, Math. Problems Eng., 2002, submitted for publication.

[25] B. Smolka, A. Chydzinski, K. Wojciechowski, K. Plataniotis, A.N. Venetsanopoulos, On the reduction of impulsive noise in multichannel image processing, Opt. Eng. 40 (6) (2001) 902–908.

[26] B. Smolka, M. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, Fast modified vector median filter, in: W. Skarbek (Ed.), Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, Vol. 2124, Springer, Berlin, 2001, pp. 570–580.

[27] B. Smolka, K. Wojciechowski, Random walk approach to image enhancement, Signal Processing 81 (3) (March 2001) 465–482.

[28] P.J. Toivanen, New geodesic distance transforms for gray scale images, Pattern Recognition Lett. 17 (1996) 437–450.

[29] P.E. Trahanias, D. Karakos, A.N. Venetsanopoulos, Directional processing of color images: theory and experimental results, IEEE Trans. Image Process. 5 (6) (1996) 868–880.

[30] P.E. Trahanias, A.N. Venetsanopoulos, Vector directional filters: a new class of multichannel image processing filters, IEEE Trans. Image Process. 2 (4) (1993) 528–534.

[31] T. Viero, K. Oistamo, Y. Neuvo, Three-dimensional median-related filters for color image sequence filtering, IEEE Trans. Circuits Systems Video Technol. 4 (2) (1994) 129–142.