Fast adaptive similarity based impulsive noise reduction filter

B. Smolka[a,*,1], R. Lukac[b,2], A. Chydzinski[a], K.N. Plataniotis[b], W. Wojciechowski[c]

[a] Department of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16 Str, 44-101, Gliwice, Poland
[b] The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto,
10 King's College Road, Toronto, Ont., Canada M5S 3G4
[c] The Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences, Baltycka Str. 5, 44-100, Gliwice, Poland

Abstract

In this paper, we address the problem of impulsive noise reduction in multichannel images. A new class of filters for noise attenuation is introduced and its relationship with commonly used filtering techniques is investigated. The computational complexity of the new filter is lower than that of the vector median filter (VMF). Extensive simulation experiments indicate that the new filter outperforms the VMF, as well as other techniques currently used to eliminate impulsive noise in color images.
© 2003 Elsevier Ltd. All rights reserved.

1. Introduction

The reduction of noise in multichannel images has been the subject of extensive research during the last years, primarily due to its importance to color image processing.

It is widely accepted, that color conveys very important information about the scene objects and this information can be used to further refine the performance of an imaging system. The most common image processing tasks are noise filtering and image enhancement. These tasks are an essential part of any image processing system, whether the final image is utilized for visual interpretation or for automatic analysis [1–3].

It has been recognized that the processing of color image data as vector fields is desirable due to the correlation that exists between the image channels, and that the nonlinear vector processing of color images is the most effective way to filter out noise. For these reasons, the new filtering technique presented in this paper is also nonlinear and utilizes the correlation among the image channels.

This paper is organized as follows. In Section 2, an overview of the standard noise reduction operations for multichannel images based on the reduced ordering of vectors is given and in Section 3 the noise models used

for the evaluation of the new filter performance are presented. Section 4 introduces a basic, non-adaptive version of the new algorithm for image enhancement. In Section 5, the construction of a new adaptive filter is presented. Experimental analysis and comparisons, in terms of quality performance and computational complexity are provided in Section 6. Finally, Section 7 summarizes this paper.

2. Standard noise reduction filters

A number of nonlinear, multichannel filters, which utilize correlation among multivariate vectors, using various distance measures have been proposed in the literature [1,4–8]. The most popular nonlinear, multichannel filters are based on the ordering of vectors in a predefined moving window. The output of these filters is defined as the lowest ranked vector according to a specific ordering technique.

Let us assume that $\mathbf{F}(x)$ represents a multichannel image and W is a window of finite size $n+1$ (filter length). The noisy image vectors inside the window W will be denoted as $\mathbf{F}_j, j = 0, 1, \ldots, n$. If the distance between two vectors $\mathbf{F}_i, \mathbf{F}_j$ is denoted as $\rho(\mathbf{F}_i, \mathbf{F}_j)$, then the scalar quantity

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

is the aggregated distance associated with the noisy vector $\mathbf{F}_i$ inside the processing window W. In the ranked order processing of multichannel signals, it is assumed that the reduced ordering of the R_i's

$$R_{(0)} \leqslant R_{(1)} \leqslant \cdots \leqslant R_{(n)}, \tag{2}$$

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

$$\mathbf{F}_{(0)}, \mathbf{F}_{(1)}, \ldots, \mathbf{F}_{(n)}. \tag{3}$$

Nonlinear ranked type multichannel estimators define the vector $\mathbf{F}_{(0)}$ as the filter output. This selection is due to the fact that vectors that diverge greatly from the data population usually appear in higher indexed locations in the ordered sequence. However, the concept of input ordering, initially applied to scalar quantities is not easily extended to multichannel data, since there is no universal way to define ordering in vector spaces. To overcome this problem, different distance functions are often utilized to order vectors according to (2) and (3) [9,6].

For example, the *vector median filter* (VMF) utilizes the Euclidean distance (L_2 norm) or the city-block distance (L_1 norm) in (1) to order vectors according to their aggregated distances. The output of the VMF is the vector $\mathbf{F}_{VMF} \in W$ for which the following condition is satisfied [10,11]:

$$\sum_{j=0}^{n} \rho(\mathbf{F}_{VMF}, \mathbf{F}_j) \leqslant \sum_{j=0}^{n} \rho(\mathbf{F}_i, \mathbf{F}_j), \quad i = 0, \ldots, n. \tag{4}$$

In other words, the original value of the pixel $\mathbf{F}_0$ in the filter window W is being replaced by $\mathbf{F}_{k*}$ such that $k* = \arg \min_k R_k$, where

$$R_k = \sum_{j=0}^{n} \rho(\mathbf{F}_k, \mathbf{F}_j), \quad k = 0, 1, \ldots, n. \tag{5}$$

The output of the VMF filter clearly depends on the applied vector norm. In this paper, we make use of the Minkowski L_p norm, with Q being the number of image channels

$$\rho(\mathbf{F}_i, \mathbf{F}_j) = \left\{ \sum_{k=1}^{Q} (F_i^k - F_j^k)^p \right\}^{1/p} \tag{6}$$

and apply the norms L_1, L_2 and $L_{\max}$ ($p = 1, p = 2$ and $p = \infty$, respectively).

Another type of filters represents the *basic vector directional filter* (BVDF). BVDF is a ranked-order, nonlinear filter which parallelizes the VMF operation [12]. However, a distance criterion, different from the L_1, L_2 norms used in VMF is utilized to rank the input vectors. The angular distance criterion used in BVDF is defined as a scalar measure

$$A_i = \sum_{j=0}^{n} \alpha(\mathbf{F}_i, \mathbf{F}_j), \quad \text{with } \alpha(\mathbf{F}_i, \mathbf{F}_j) = \cos^{-1} \left(\frac{\mathbf{F}_i \cdot \mathbf{F}_j}{|\mathbf{F}_i||\mathbf{F}_j|} \right). \tag{7}$$

An ordering of the A_i's implies the same ordering to the corresponding vectors $\mathbf{F}_i$ as in (3). The BVDF outputs the vector $\mathbf{F}_{(0)}$ that minimizes the sum of angles with all the other vectors within the processing window. In other words, the BVDF chooses the vector most centrally located without considering the magnitudes of the input vectors.

To improve the efficiency of the directional filters, another method called *directional-distance filter* (DDF) was proposed in [13]. DDF constitutes a combination of VMF and VDF and is derived by attempting to minimize a combined distance measure. If $\alpha(\mathbf{F}_i, \mathbf{F}_j)$ denotes the angle between $\mathbf{F}_i$ and $\mathbf{F}_j$ and $\rho(\mathbf{F}_i, \mathbf{F}_j)$ denotes the distance between them, then

$$\Omega_i = \left[\sum_{j=0}^{n} \alpha(\mathbf{F}_i, \mathbf{F}_j) \right]^{1-\kappa} \cdot \left[\sum_{j=0}^{n} \rho(\mathbf{F}_i, \mathbf{F}_j) \right]^{\kappa}, \tag{8}$$

is assigned to the vector $\mathbf{F}_i$. If we arrange the values in ascending order $\Omega_{(0)} \leqslant \Omega_{(1)} \leqslant \cdots \leqslant \Omega_{(\tau)} \cdots \leqslant \Omega_{(n)}$, then the output of DDF is $\mathbf{F}_0$, which corresponds to $\Omega_{(0)}$. For $\kappa = 0$, we obtain the BVDF and for $\kappa = 1$ the VMF. The DDF is defined for $\kappa = 0.5$ and its usefulness stems from the fact that it combines both the criteria used in BVDF and VMF [5,10,11,14].

3. Noise models

For the evaluation of the new class of noise suppression algorithms, six types of impulsive noise were used to simulate different types of distortions which may corrupt color images.

(I) *Impulsive noise*:

(a) Let $\mathbf{F} = \{F_R, F_G, F_B\}$ denote the original pixel and let $\mathbf{F}'$ denote the pixel corrupted by the noise process. Then the image pixels are distorted according to the following scheme:

$$\mathbf{F}' = \begin{cases} \{d_1, F_G, F_B\} & \text{with probability } p, \\ \{F_R, d_2, F_B\} & \text{with probability } p, \\ \{F_R, F_G, d_3\} & \text{with probability } p. \end{cases} \tag{9}$$

where d_1, d_2, d_3 are independent and equal to 0 or 255 with equal probability.

(b) In this noise model, the RGB channels are corrupted with impulsive noise (0 or 255) with probability p like in (a), but the contamination process is correlated at level r, which means that if one of the image channels has been corrupted, then the probability that an additional channel will be corrupted is equal to r.

(II) *Uniform noise*: $\mathbf{F}' = \{d_1, d_2, d_3\}$ with probability p, where $d_1, d_2, d_3 \in [0, 255]$ are uniformly distributed independent integer values.

(III) *Impulsive noise in HSV color space*: The third noise model is signal dependent and operates in the HSV color space. The noisy image pixel is generated

according to the following rule:

$$\mathbf{F}' = \begin{cases} \{H, S, V\} & \text{with probability } (1-p), \\ \{\delta, S, V\} & \text{with probability } p_1 p, \\ \{H, \delta, V\} & \text{with probability } p_2 p, \\ \{H, S, \delta\} & \text{with probability } p_3 p, \\ \{\delta, \delta, \delta\} & \text{with probability } p_4 p, \end{cases} \qquad (10)$$

where p denotes the degree of impulsive noise distortion, $\sum_i p_i = 1, i = 1, \ldots, 4$, and δ is a positive or negative impulse. In this noise model, δ is a random variable in a small range very close to the upper or lower bound of a HSV component, according to the original value of this component [15].

The *root mean squared error* (*RMSE*), *peak signal to noise ratio* (*PSNR*), *normalized mean square error* (*NMSE*) [3] were used for the analysis of the efficiency of the proposed filter class.

The objective quality measures are defined by the following formulas:

RMSE

$$= \sqrt{\frac{1}{NMQ} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{q=1}^{Q} (F^q(i,j) - \hat{F}^q(i,j))^2},$$

$$PSNR = 20 \log \left(\frac{255}{RMSE} \right), \qquad (11)$$

$$NMSE = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{q=1}^{Q} (F^q(i,j) - \hat{F}^q(i,j))^2}{\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{q=1}^{Q} F^q(i,j)^2}, \qquad (12)$$

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

4. Proposed algorithm

4.1. Gray-scale images

Let us assume a filtering window W containing $n+1$ image pixels $\{F_0, \ldots, F_n\}$, where n is the number of neighbors of the central pixel F_0 (see Fig. 1a) and let us define the similarity function $\mu : [0; \infty) \rightarrow \mathbf{R}$ which is non-ascending and convex in $[0; \infty)$ and satisfies $\mu(0) = 1, \mu(\infty) = 0$. The similarity between two pixels of the same intensity should be 1, and the similarity between pixels with minimal and maximal gray-scale values should be very close to 0. The function $\mu(F_i, F_j)$ defined as $\mu(F_i, F_j) = \mu(|F_i - F_j|)$ can easily satisfy the required conditions.

Let us additionally define the cumulated sum M of similarities between a given pixel and all other pixels belonging to window W. For the central pixel, we introduce M_0 and for the neighbors of F_0 we define
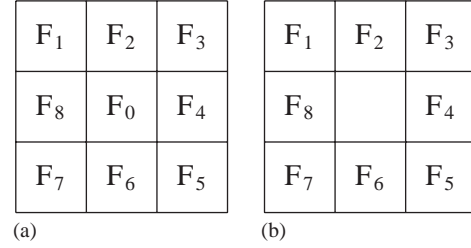

Fig. 1. Illustration of the construction of the new filtering technique. First the cumulative similarity value M_0 between the central pixel F_0 and its neighbors is calculated (a), then pixel F_0 is rejected from the filter window and the cumulative similarity values $M_k, k = 1, \ldots, n$ of the pixels $F_1, \ldots, F_n$ are determined (b).

M_k as

$$M_0 = \sum_{j=1}^{n} \mu(F_0, F_j), \quad M_k = \sum_{\substack{j=1 \\ j \neq k}}^{n} \mu(F_K, F_j), \qquad (13)$$

which means that for F_k which are neighbors of F_0 we do not take into account the similarity between F_k and F_0, which is the main idea behind the new algorithm. The omission of the similarity $\mu(F_k, F_0)$ when calculating M_k, privileges the central pixel, as in the calculation of M_0 we have n similarities $\mu(F_0, F_k), k = 1, 2, \ldots, n$ and for $M_k, k > 0$ we have only $n - 1$ similarity values, as the central pixel F_0 is excluded from the calculation of the sum M_k (Fig. 1b).

In the construction of the new filter, the reference pixel F_0 in the window W is replaced by one of its neighbors if $M_0 < M_k, k = 1, \ldots, n$. If this is the case, then F_0 is replaced by that F_{k*} for which $k* = \arg \max M_k, k = 1, \ldots, n$.

In other words, F_0 is detected as being corrupted if $M_0 < M_k, k = 1, \ldots, n$ and is replaced by its neighbors F_k which maximizes the sum of similarities M between all the pixels of W excluding the central pixel.

The basic assumption is that a new pixel must be taken from the window W (introducing pixels which do not occur in the image is prohibited like in the VMF). For this purpose μ must be convex, which means that in order to find a maximum of the sum of similarity functions M it is sufficient to calculate the values of M only in points $F_0, F_1, \ldots, F_n$ [16,17].

4.2. Multichannel images

The presented approach can be applied in a straightforward way to multichannel images. We use the similarity function defined by $\mu\{\mathbf{F}_i, \mathbf{F}_j\} = \mu(\|\mathbf{F}_i - \mathbf{F}_j\|)$, where $\|\cdot\|$ denotes the specific vector norm. Now in exactly the same way we can maximize the total similarity function M for the vector case.

We have checked several convex functions in order to compare our approach with the standard filters used in color image processing presented in Table 1 and we have

obtained good results (Table 2), when applying the following similarity functions, which can be treated as kernels of non-parametric density estimation [22,23]:

$$\mu_0(x) = \exp\left\{-\left(\frac{x}{h}\right)^2\right\}, \quad \mu_1(x) = \exp\left\{-\frac{x}{h}\right\},$$

$$\mu_2(x) = \frac{1}{1 + x/h}, \qquad h \in (0; \infty),$$

$$\mu_3(x) = \frac{1}{(1+x)^h}, \quad \mu_4(x) = 1 - \frac{2}{\pi}\arctan\left(\frac{x}{h}\right),$$

$$\mu_5(x) = \frac{2}{1 + \exp\{x/h\}}, \qquad h \in (0; \infty),$$

$$\mu_6(x) = \frac{1}{1 + x^h}, \quad \mu_7(x) = \begin{cases} 1 - x/h & \text{if } x \leqslant h, \\ 0 & \text{if } x > h, \end{cases}$$

$$h \in (0; \infty).$$

It is interesting to note, that very good results were achieved for the simplest similarity function $\mu_7(x)$, (see Fig. 2, Table 2), which allows to construct a fast noise reduction algorithm. In the multichannel case, we have

$$M_0 = \sum_{j=1}^{n} \mu(\mathbf{F}_0, \mathbf{F}_j), \quad M_k = \sum_{\substack{j=1 \\ j \neq k}}^{n} \mu(\mathbf{F}_k, \mathbf{F}_j), \tag{14}$$

where $\rho\{\mathbf{F}_i, \mathbf{F}_k\} = \|\mathbf{F}_k - \mathbf{F}_l\|$ and $\|\cdot\|$ is the L_2 vector norm, as it yields the best results (Tables 3 and 4). Applying the linear similarity function μ_7 we obtain

$$\mu(\mathbf{F}_i, \mathbf{F}_k) = \begin{cases} 1 - \dfrac{\rho(\mathbf{F}_i, \mathbf{F}_k)}{h} & \text{for } \rho(\mathbf{F}_i, \mathbf{F}_k) < h, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Table 1
Filters taken for comparison with the proposed adaptive filter

Notation	Filter
AMF	Arithmetic mean [3]
VMF	Vector median [18]
BVDF	Basic vector directional [12]
GVDF	Generalized vector directional [13]
DDF	Directional-distance [4]
HDF	Hybrid directional [19]
AHDF	Adapt. hybrid directional [19]
FVDF	Fuzzy vector directional [20]
ANNF	Adapt. nearest neighbor [21]

Table 2
Comparison of the new algorithm based on different kernel functions with the standard techniques, using the LENA color image contaminated by 5% of impulsive noise (model II).

Method	NMSE (10^{-4})	RMSE	PSNR (dB)
AMF	82.863	12.903	25.917
VMF	23.304	6.842	31.427
BVDF	29.074	7.643	30.466
DDF	24.003	6.944	31.288
HDF	22.845	6.775	31.513
AHDF	22.603	6.739	31.559
FVDF	26.755	7.331	30.827
ANNF	31.271	7.926	30.149
Filtering kernels			
$\mu_0(x)$	5.056	3.163	38.137
$\mu_1(x)$	4.959	3.157	38.145
$\mu_2(x)$	5.398	3.294	37.776
$\mu_3(x)$	9.574	4.387	35.288
$\mu_4(x)$	5.064	3.190	38.054
$\mu_5(x)$	4.777	3.099	38.307
$\mu_6(x)$	11.024	4.707	34.675
$\mu_7(x)$	4.693	3.072	38.384

Table 3
Results of the new adaptive algorithm for the LENA image (noise model I(b)) with $p = 0.04$. The vector median with L_2 norm gave $PSNR = 32.4$

Norm	NMSE	RMSE	PSNR	h
L_1	5.042	3.183	38.074	6.580
L_2	4.659	3.060	38.417	6.358
L_∞	5.304	3.265	37.854	6.505

Table 4
Adaptive algorithm results for the image PEPPERS (noise model I(b)) with $p = 0.04$. The VMF with L_2 norm gave $PSNR = 31.6$

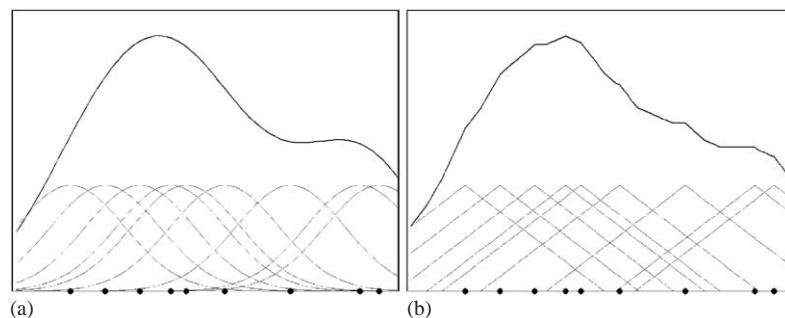| Norm	NMSE	RMSE	PSNR	h
L_1	9.236	3.888	36.337	10.137
L_2	8.426	3.713	36.736	9.366
L_∞	9.960	4.038	36.008	9.236

(a)   (b)

Fig. 2. Cumulative similarity values dependence on the pixel gray-scale value for a window containing a set of pixels with intensities 15, 24, 33, 41, 45, 55, 72, 90, 95 using the μ_0 function (a) and μ_7 function (b).

Then we have from (14)

$$M_0 = n - \frac{1}{h} \sum_{j=1}^{n} \rho(\mathbf{F}_0, \mathbf{F}_j) \quad \text{and}$$

$$M_k = \sum_{\substack{j=1, \\ j \neq k}}^{n} \left(1 - \frac{\rho(\mathbf{F}_k, \mathbf{F}_j)}{h} \right)$$

$$= n - 1 - \frac{1}{h} \sum_{j=1}^{n} \rho(\mathbf{F}_k, \mathbf{F}_j). \tag{16}$$

In this way, the difference between M_0 and M_k is

$$M_0 - M_k = n - \frac{1}{h} \sum_{j=1}^{n} \rho(\mathbf{F}_0, \mathbf{F}_j)$$

$$- \left[n - 1 - \frac{1}{h} \sum_{j=1}^{n} \rho(\mathbf{F}_k, \mathbf{F}_j) \right]$$

$$= 1 - \frac{1}{h} \sum_{j=1}^{n} [\rho(\mathbf{F}_0, \mathbf{F}_j) - \rho(\mathbf{F}_k, \mathbf{F}_j)], \tag{17}$$

$$M_0 - M_k > 0 \quad \text{if } h > \sum_{j=1}^{n} [\rho(\mathbf{F}_0, \mathbf{F}_j) - \rho(\mathbf{F}_k, \mathbf{F}_j)],$$

$$k = 1, \dots, n. \tag{18}$$

If this condition is satisfied, then the central pixel is considered as not disturbed by the noise process, otherwise the pixel $\mathbf{F}_k$ for which the cumulative similarity value achieves maximum, replaces the central noisy pixel. In this way, the filter changes the central pixel only when it is really noisy and preserves the original undistorted image structures.

The construction of the new filter is presented in Fig. 3 for the one-dimensional case (gray-scale images). In the example shown in this figure, the supporting window W of size 3×3 contains 9 pixels of intensities $15, 24, 33, 41, 45, 55, 72, 90, 95$ (their special arrangement is not relevant). Each of the graphs from (a) to (i) shows the dependence of M_0 and $M_{/0}$ on the gray-scale value ($M_{/0} < M_0$), where $M_{/0}$ denotes the cumulative similarity value with rejected central pixel F_0 on the gray-scale value. Graph (a) shows the plot of M_0 and $M_{/0}$ for $F_0 = 15$, plot (b) for $F_0 = 24$ and so on till plot (i) which shows the graphs of M_0 and $M_{/0}$ for $F_0 = 95$. The central pixel will be replaced in cases: (a), (b), (f)–(i), as in those cases there exists a pixel F_k for which $M_0 < M_k$ or $R_0 > R_k$. Note, that actually only the values in points $F_k, k = 0, \dots, n$ are of importance; however, the continuous plot shows that the extremum of the similarity function $M_{/0}$ is always obtained in points F_k, which is an
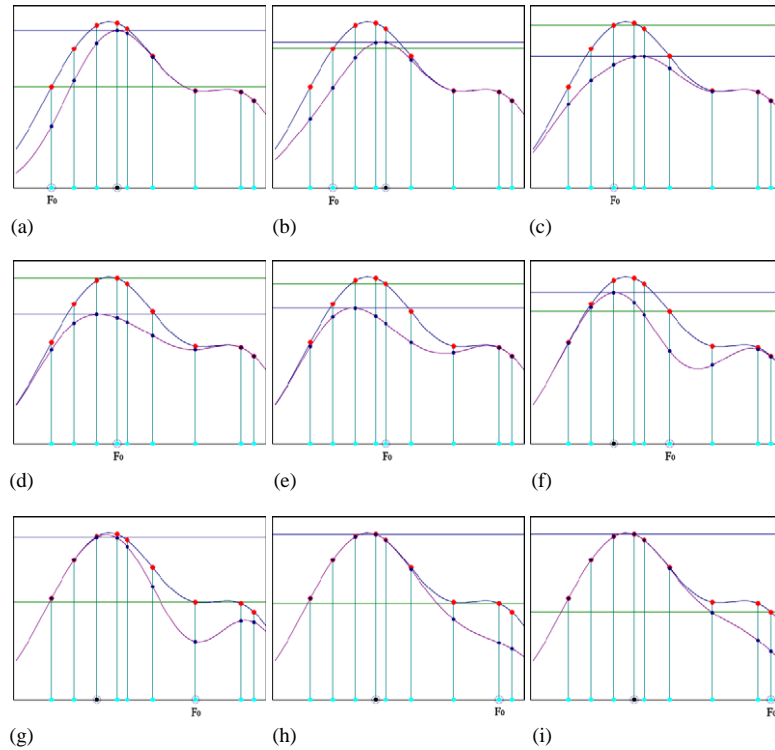

Fig. 3. Illustration of the new filter construction using the Gaussian kernel (gray-scale case). The supporting window W of size 3×3 contains 9 pixels of intensities 15, 24, 33, 41, 45, 55, 72, 90, 95. Each of the graphs from (a) to (i) shows the dependence of M_0 and $M_{/0}$, ($M_{/0} < M_0$), where $M_{/0}$ denotes the cumulative similarity value with rejected central pixels F_0 on the gray-scale value. Graph (a) shows the plot of M_0 and $M_{/0}$ for $F_0 = 15$, plot (b) for $F_0 = 24$ and so on till plot (i) which shows the graphs of M_0 and $M_{/0}$ for $F_0 = 95$. The arrangement of pixels surrounding the central pixel F_0 is not relevant. The central pixel will be replaced in cases: (a), (b), (f)–(i), as in those cases there exists a pixel for which $M_0 < M_k$ or $R_0 > R_k$.
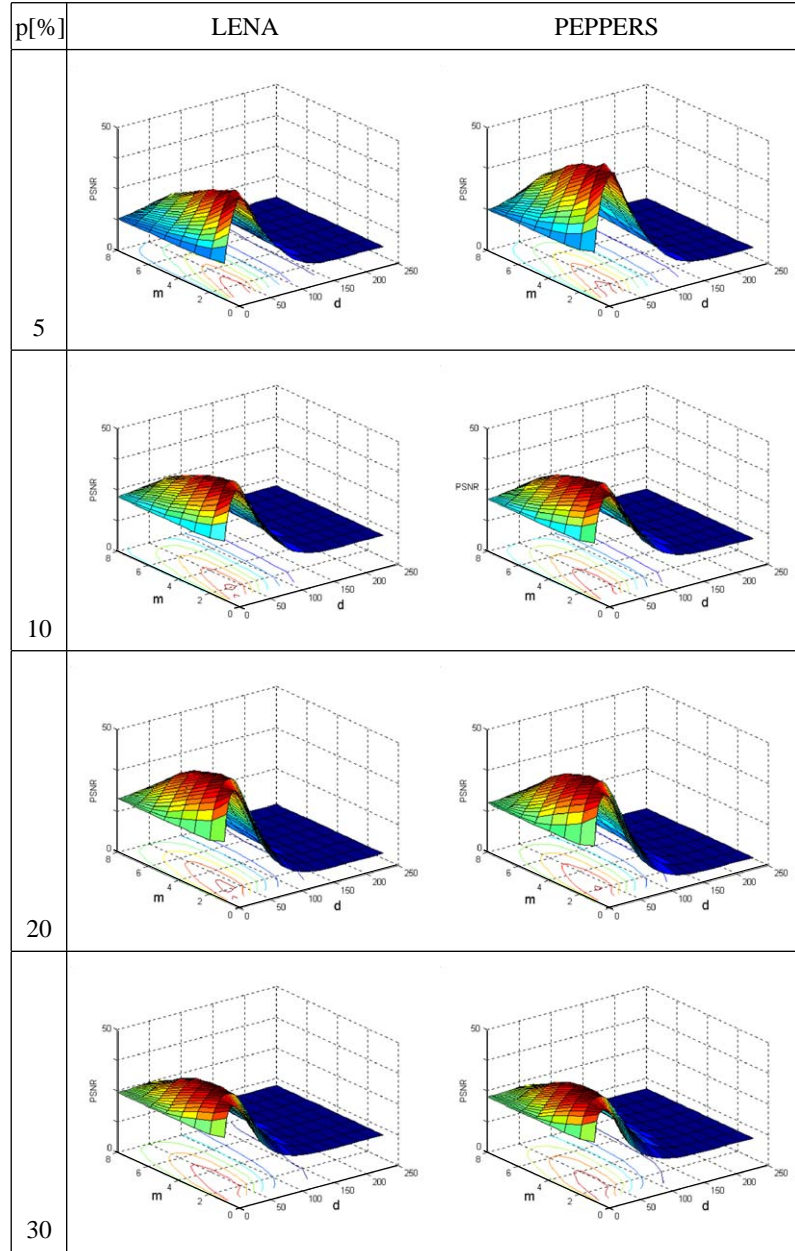
p[%]	LENA	PEPPERS
5		
10		
20		
30		

Fig. 4. Dependence of the efficiency of the proposed algorithm in terms of *PSNR* for different number of 'close' neighbors m and the threshold distance d. For the evaluation color test image LENA, PEPPERS and GOLDHILL contaminated with impulsive noise of model II were used. As can be seen good results are obtained for $m = 2$ and $d \in [40, 60]$.

important feature of this algorithm. Because the function $M_{/0}$ is convex, therefore the maximum can be found by calculating the similarity values without F_0 in n points only, which makes the algorithm computationally attractive.

It is easy to observe that the construction of the new filter is quite similar to the standard VMF. Instead of the original function R_k in (1), a modified cumulative distance function R is proposed:

$$R_k = \begin{cases} -h + \sum_{j=1}^n \rho(\mathbf{F}_k, \mathbf{F}_j), & \text{for } k = 0, \\ \sum_{j=1}^n \rho(\mathbf{F}_k, \mathbf{F}_j), & \text{for } k = 1, \ldots, n, \end{cases} \quad (19)$$

and in the same way as in VMF, the original vector $\mathbf{F}_0$ in the filter window W is being replaced by $\mathbf{F}_{k*}$ such that $k^* = \arg \min_k R_k$.

It is easy to notice that the above construction is equivalent to the condition expressed in (18). Now, instead of maximizing the cumulative similarity M_k, we minimize the modified cumulative distance R_k. In this way, the condition for retaining the original image pixel is: $R_0 < R_k, k = 1, \ldots, n$, which leads to the condition

$$-h + \sum_{j=1}^n \rho(\mathbf{F}_0, \mathbf{F}_j) \leqslant \sum_{j=1}^n \rho(\mathbf{F}_k, \mathbf{F}_j), \quad k = 1, \ldots, n, \quad (20)$$
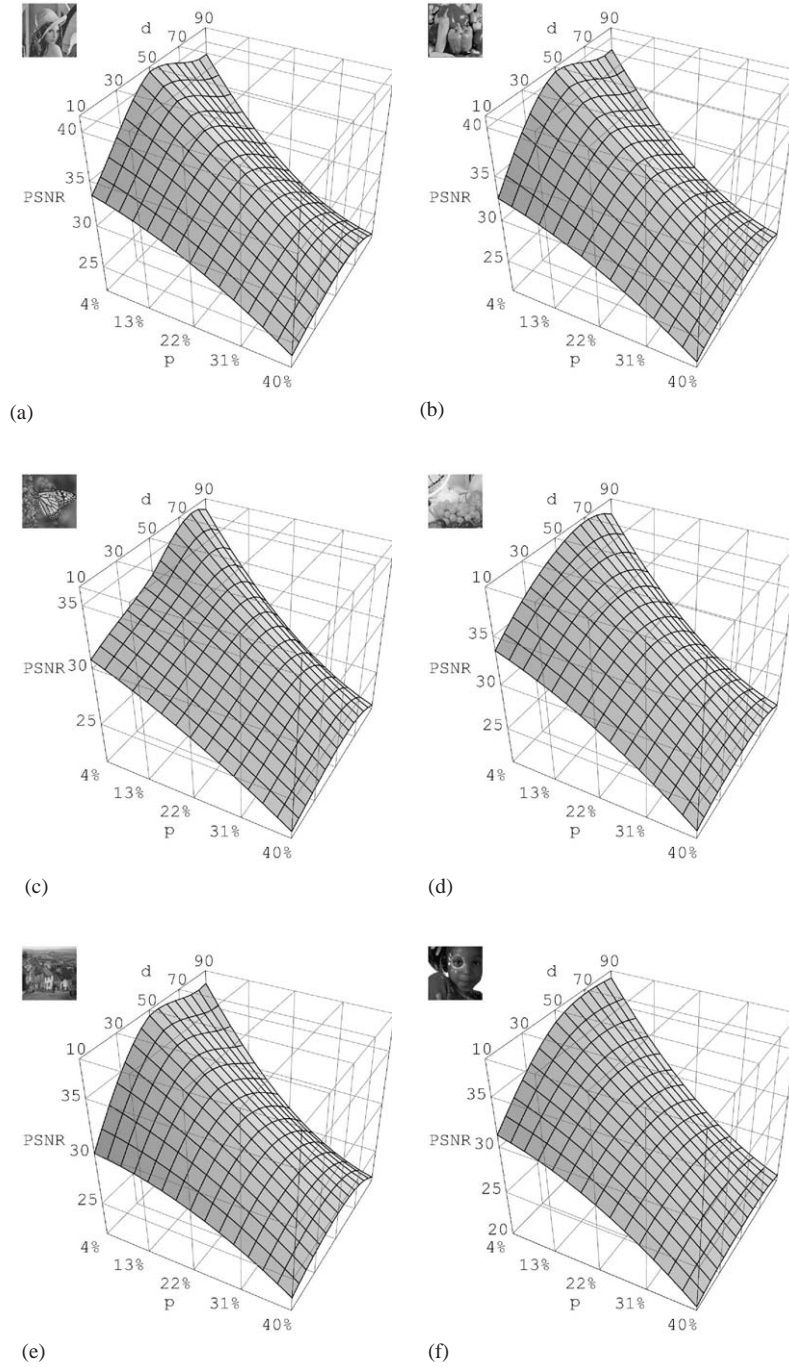
Fig. 5. Dependence of the efficiency of the proposed algorithm in terms of *PSNR* for the number of 'close' neighbors $m = 2$ on the threshold distance d for color test images LENA, PEPPERS, MONARCH, FRUITS, GOLDHILL, GIRL ((a)–(f)), contaminated with impulsive noise of model II.

$$R_0 \leqslant R_k \quad \text{if } h \geqslant \sum_{j=1}^{n} [\rho(\mathbf{F}_0, \mathbf{F}_j) - \rho(\mathbf{F}_k, \mathbf{F}_j)],$$

$$k = 1, \ldots, n. \tag{21}$$

The construction of the new filter is similar to that of VMF; however, the major difference is the omitting of the central pixel $\mathbf{F}_0$, when calculating $R_k, k > 0$. This scheme, based on the simple *leave-one-out* scheme, is the *most important feature* of the new algorithm. As the central pixel is suspected to be noisy, it is not taken into consideration, when calculating the distances associated with the neighbors of $\mathbf{F}_0$. In this way, the filter replaces the central pixel only when it is really noisy, while preserving the original undisturbed image structures.

As it can be easily seen, the parameter h in (18) and (19) influences the intensity of the filtration process. In

other words, the fraction of pixels replaced by the new filter is a decreasing function of h (see Figs. 7 and 9).

For $h = 0$ this fraction is close to that one caused by VMF. For $h \to \infty$, this fraction is decreasing to zero (there is no filtering at all).

As widely known, VMF has the disadvantage of replacing too many uncorrupted image pixels. It is improved in the new filter design by setting positive h values, which forces the filter to preserve uncorrupted,
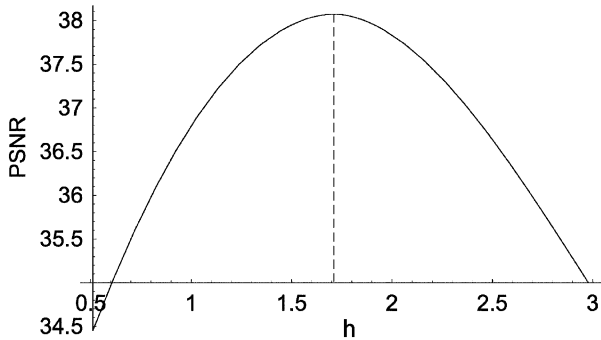

Fig. 6. Dependence of the filtering results on h (LENA image with 11.5% of corrupted pixels and L_1 norm, noise model I(b) with $p = 0.04$). The VMF output gave $PSNR = 32.4$.
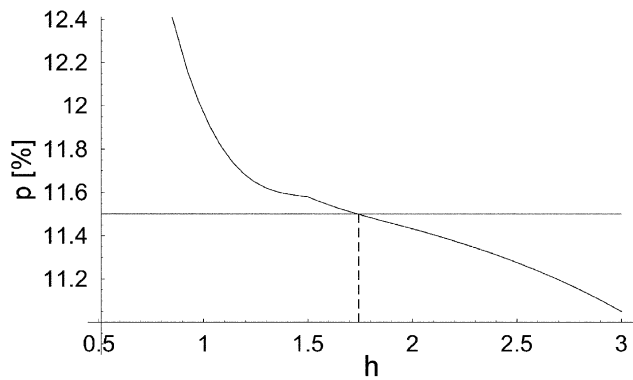

Fig. 7. Dependence of the fraction of pixels replaced by the filter on the h value (LENA image with 11.5% of pixels corrupted, noise model I(b) with $p = 0.04$ and L_1 norm).

Table 5
The comparison of real and estimated fractions of the noisy pixels for $d = 50$.

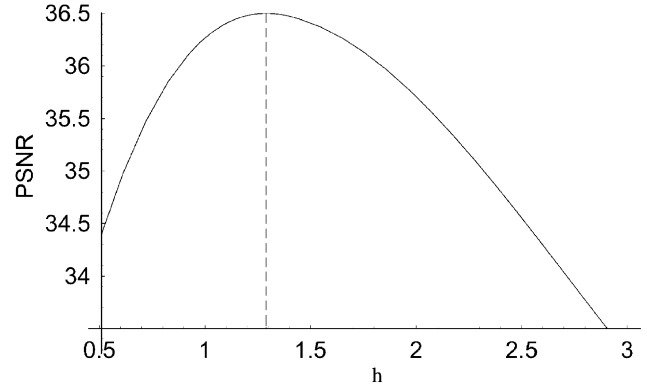| Real p	Estimated p (LENA)	Estimated p (PEPPERS)
0.01	0.0113	0.0122
0.02	0.0206	0.0216
0.05	0.0500	0.0510
0.10	0.0980	0.0986
0.20	0.1942	0.1964
0.40	0.3972	0.3973
0.70	0.7501	0.7504

Fig. 8. Dependence of the results of the filtering on h (PEPPERS image with 11.5% of pixels corrupted, noise model I(b) with $p = 0.04$ and L_1 norm). The VMF output gave $PSNR = 31.6$.
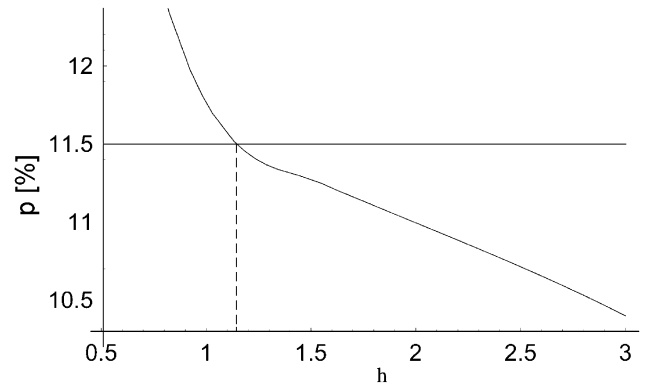

Fig. 9. Dependence of the fraction of pixels replaced by the filter on h (PEPPERS image with 11.5% of pixels, noise model I(b) with $p = 0.04$ and L_1 norm).

Table 6
The comparison of real and estimated fractions of the noisy pixels for variants B and C of the estimator, LENA image.

Real p	Estimated p variant B	Estimated p variant C
0.01	0.0099	0.0158
0.02	0.0192	0.0253
0.05	0.0476	0.0547
0.10	0.0933	0.0103
0.20	0.1821	0.2016
0.40	0.3541	0.4301
0.70	0.5981	0.8472

original pixels, but still enables to remove corrupted ones. Moreover, h can be controlled for the best effectiveness of the filter depending on image structure and noise statistics. The subject of setting the value of h will be addressed in detail in the next section.

It is easy to observe that the presented method is faster than VMF. It can be shown using a simple matrix representation (for the sake of simplicity in the four-neighborhood system case). In order to find R_{k*} using

Table 7
Comparison of the efficiency of the new algorithm with the standard techniques (Table 1) using LENA standard image and different vector norms (noise model I(b)) with $p = 0.04$, precisely 11.5% of the pixels have been corrupted with impulsive noise

Filter	NMSE (10^{-4})	RMSE	PSNR (dB)
AMF	79.317	12.627	26.105
VMF	18.766	6.142	32.365
BVDF	24.587	7.030	31.192
DDF	18.872	6.159	32.340
HDF	18.610	6.116	32.401
AHDF	18.310	6.067	32.472
FVDF	22.251	6.688	31.625
ANNF	26.800	7.340	30.817
NEW	4.659	3.060	38.417

Table 8
Comparison of the efficiency of the new algorithm with the standard techniques (Table 1) using PEPPERS standard image and different vector norms (noise model—I(b)) with $p = 0.04$, precisely 11.5% of the pixels have been corrupted with impulsive noise

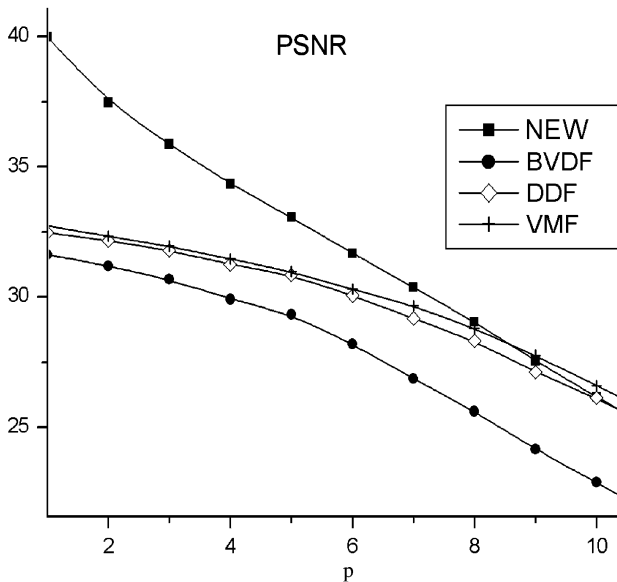| Filter	NMSE (10^{-4})	RMSE	PSNR (dB)
AMF	108.650	13.338	25.629
VMF	27.570	6.719	31.585
BVDF	47.944	8.860	29.182
DDF	28.179	6.793	31.490
HDF	26.819	6.627	31.705
AHDF	26.430	6.579	31.768
FVDF	33.337	7.388	30.760
ANNF	45.115	8.595	29.446
NEW	8.426	3.713	36.736

Fig. 10. The efficiency of the new adaptive algorithm in terms of *PSNR* value in comparison with standard noise reduction filters. The test color image LENA has been contaminated by noise process I(a) with p from 1% to 10%.

the VMF method we have to add the elements in rows or columns of the following symmetric matrix:

$$\mathbf{T}_{VMF} = \begin{bmatrix} 0 & \rho_{01} & \rho_{02} & \rho_{03} & \rho_{04} \\ \rho_{10} & 0 & \rho_{12} & \rho_{13} & \rho_{14} \\ \rho_{20} & \rho_{21} & 0 & \rho_{23} & \rho_{24} \\ \rho_{30} & \rho_{31} & \rho_{32} & 0 & \rho_{34} \\ \rho_{40} & \rho_{41} & \rho_{42} & \rho_{43} & 0 \end{bmatrix},$$

where $\rho_{ij} = \rho_{ji} = \rho(\mathbf{F}_i, \mathbf{F}_j)$. (22)

Obviously, the symmetry of the matrix $\mathbf{T}_{VMF}$ causes that effectively we have to compute 10 distances. (36 in the eight-neighborhood case) and then to make 15 additions (63 in the eight-neighborhood case). The appropriate matrix for the new filter has the
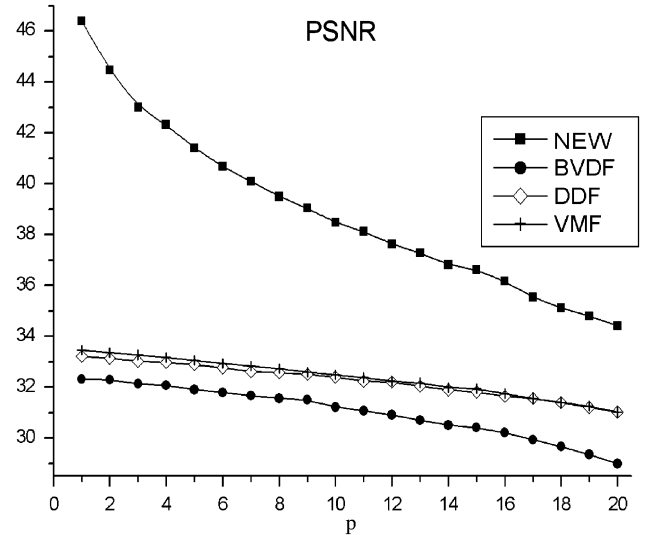

Fig. 11. The efficiency of the new adaptive algorithm in terms of *PSNR* value in comparison with the standard noise reduction filters. The test color image LENA has been contaminated by noise process II with p from 1% to 20%.
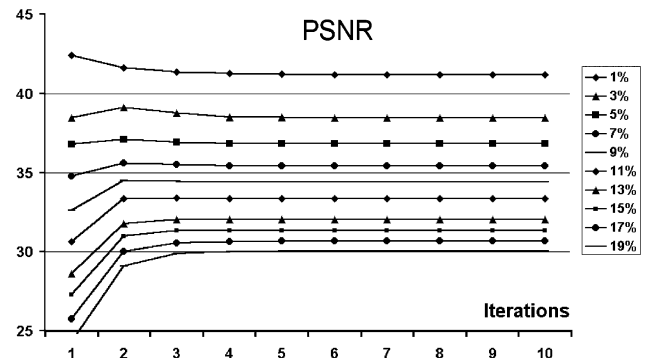

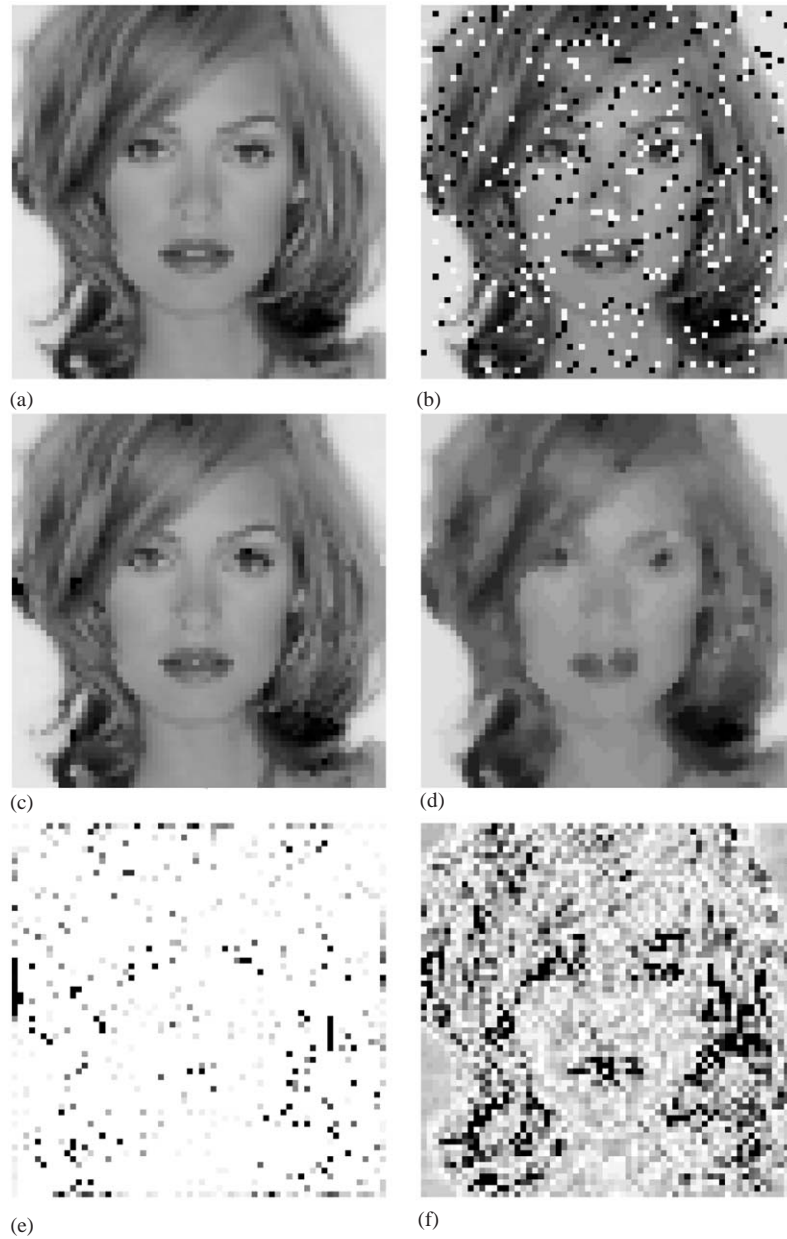Fig. 12. Dependence of the *PSNR* on the number of filtering iterations (LENA noise model I(b) with $p = 0.04$).

Fig. 13. Comparison of the efficiency of the vector median and the proposed filter: (a) test image, (b) image contaminated by 2% impulse noise I(a), (c) image filtered using the proposed technique, (d) output of the vector median, (e) and (f) the absolute difference between the original and filtered image, for both the new method (left) and the vector median (right) (the absolute difference of the gray-scale values were multiplied by factor 5 in order to better visualize the detail preservation property of the new filter).

form:

$$\mathbf{T}_{NEW} = \begin{bmatrix} -h & \rho_{01} & \rho_{02} & \rho_{03} & \rho_{04} \\ 0 & 0 & \rho_{12} & \rho_{13} & \rho_{14} \\ 0 & \rho_{21} & 0 & \rho_{23} & \rho_{24} \\ 0 & \rho_{31} & \rho_{32} & 0 & \rho_{34} \\ 0 & \rho_{41} & \rho_{42} & \rho_{43} & 0 \end{bmatrix}. \quad (23)$$

The number of distances we need is still 10 but there are only 11 additions (55 in the eight-neighborhood case), so the new filter is faster than VMF and it also outperforms

VMF in terms of the objective quality measures, as will be shown in Section 6.

5. Adaptive filter design

To enhance the performance of the proposed filter, the parameter h can be determined in an adaptive way depending on the image structure, properties and intensity of the noise process.

Experiments performed on color images LENA and PEPPERS indicate that the $PSNR$ reaches its maximum
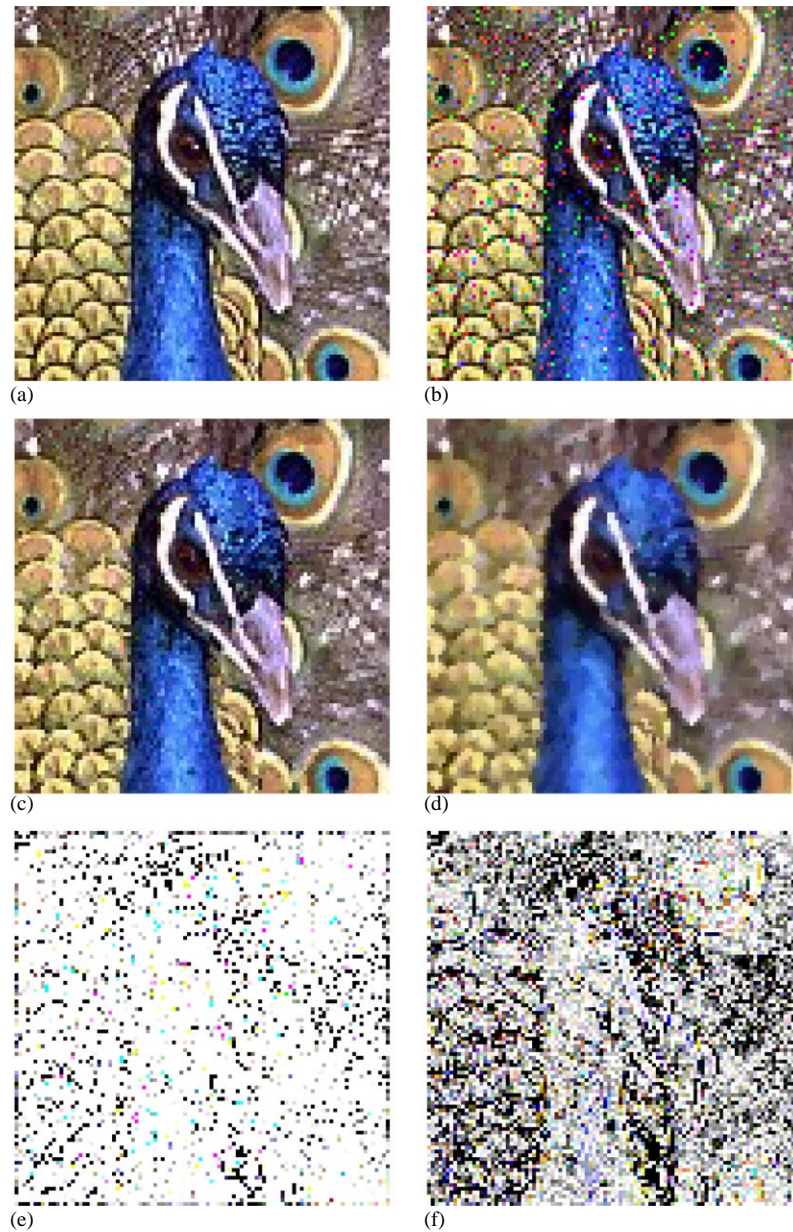
Fig. 14. Comparison of the efficiency of the vector median and the proposed filter: (a) test image, (b) image contaminated by 2% impulse noise I(a), (c) image filtered using the proposed technique, (d) output of the vector median, (e) and (f) the absolute difference between the original and filtered image, for both the new method (left) and the vector median (right) (the RGB values were multiplied by factor 5 in order to enhance the difference).

for that value of the parameter h, that leads to a number of pixel replacements equal to the number of noisy pixels in the noise-corrupted image. Figs. 6 and 7 and Figs. 8 and 9 which show the performance of the new filter as a function of h and depict the fraction of pixels replaced as a function of h, validate this observation. This observation suggests that superior filtering results can be obtained by adapting the h value to the image structure and noise statistics.

In this way, the filtering process proceeds as follows:

1. Estimation of the fraction of corrupted pixels.

2. Finding optimal value of h.
3. Final filtering using the obtained optimal value of h.

In most applications, the noise intensity is unknown and we need to find a robust estimator of the fraction of corrupted pixels. In this paper, the following estimator is applied:

In the analysis of all the pixels which build an image, a pixel is considered to be undamaged by the noise process, if among eight of its neighbors, there exist at least m pixels which, are 'close' to it.

Two pixels are declared to be 'close' if the L_2 distance between them, in the RGB color space, is less than a
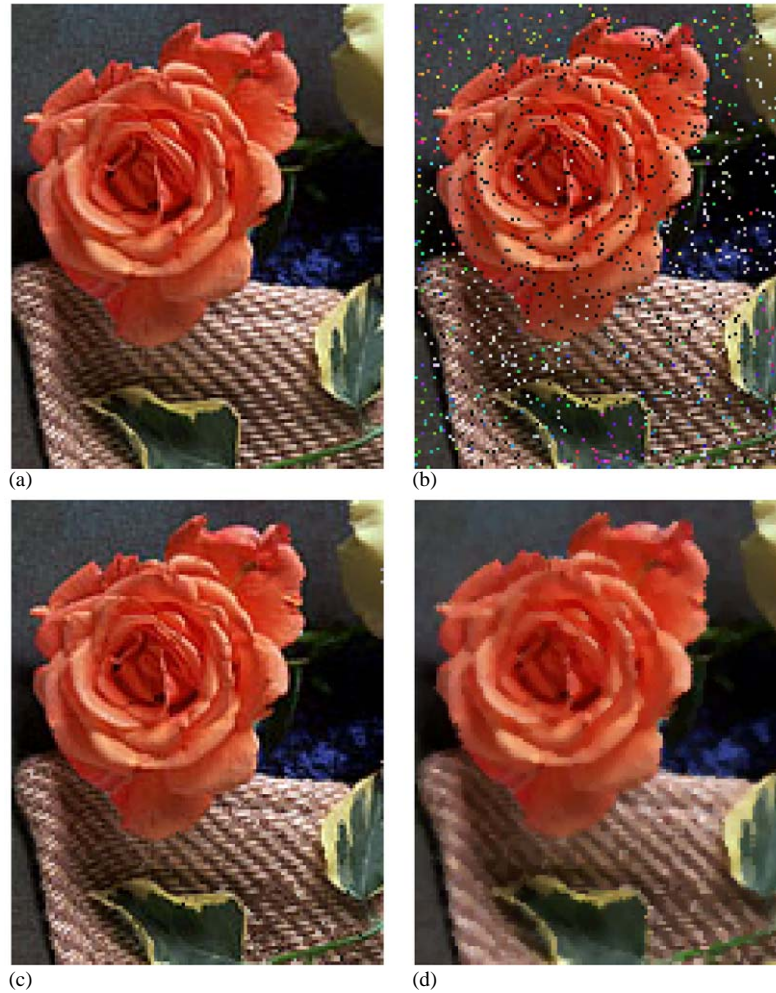
Fig. 15. Comparison of the efficiency of the vector median and the proposed filter: (a) test image ROSE, (b) image contaminated by 6% impulsive noise (III, $p = 0.06, p_1 = p_2 = p_3 = 0, p_s = 1$)—[15], (c) image filtered using the proposed adaptive technique, (d) output of the vector median with L_2 norm.

predefined constant d. As has been experimentally evaluated, this estimator works correctly, even for images with completely different structures.

Table 5 shows the result of estimation of p using the described estimator for two test color images LENA and PEPPERS and different fractions of the corrupted pixels p.

The value of the critical distance d used in the construction of the estimator is not critical, values of d in the range [40,60] give acceptable results. Fig. 4 shows the dependence of the PSNR for color test LENA and PEPPERS, contaminated by noise model II on the m and d parameters. As can be seen, good results are obtained for $m = 2$ and $d \in [40, 60]$. This is also confirmed by Fig. 5, which presents the filtering efficiency dependence on the threshold d for $m = 2$.

One can also use such estimators as: a pixel is considered to be undamaged, if among eight of its neighbors, there exist at least one ($m = 1$) which is 'close' to it (variant B), or a pixel is considered to be undamaged by the noise process, if among eight of its

neighbors, there exist at least three ($m = 3$) which are 'close' to it, (variant C). These variants also produce acceptable results (see Table 6), but for obvious reasons variant B has the tendency to underestimate, while variant C to overestimate the impulsive noise fraction. It is also easy to observe that the value of $m = 2$ preserves lines and corners, and therefore we used this parameter for the estimation purposes.

As regards point 2, the constant h has to be set for the value, for which the percentage of pixels changed by the new filter is equal to the estimated noise fraction p. In order to design a fast filter implementation, the well-known method of bisection can be used. This method allows to find the root of an equation $g(x) = 0$ in $[a, b]$ providing that $g(x)$ is continuous and $g(a)g(b) < 0$. In the case considered here,

$$g(h) = \gamma(h) - p, \tag{24}$$

where $\gamma(h)$ is the fraction of pixels changed by the filter, depending on h.
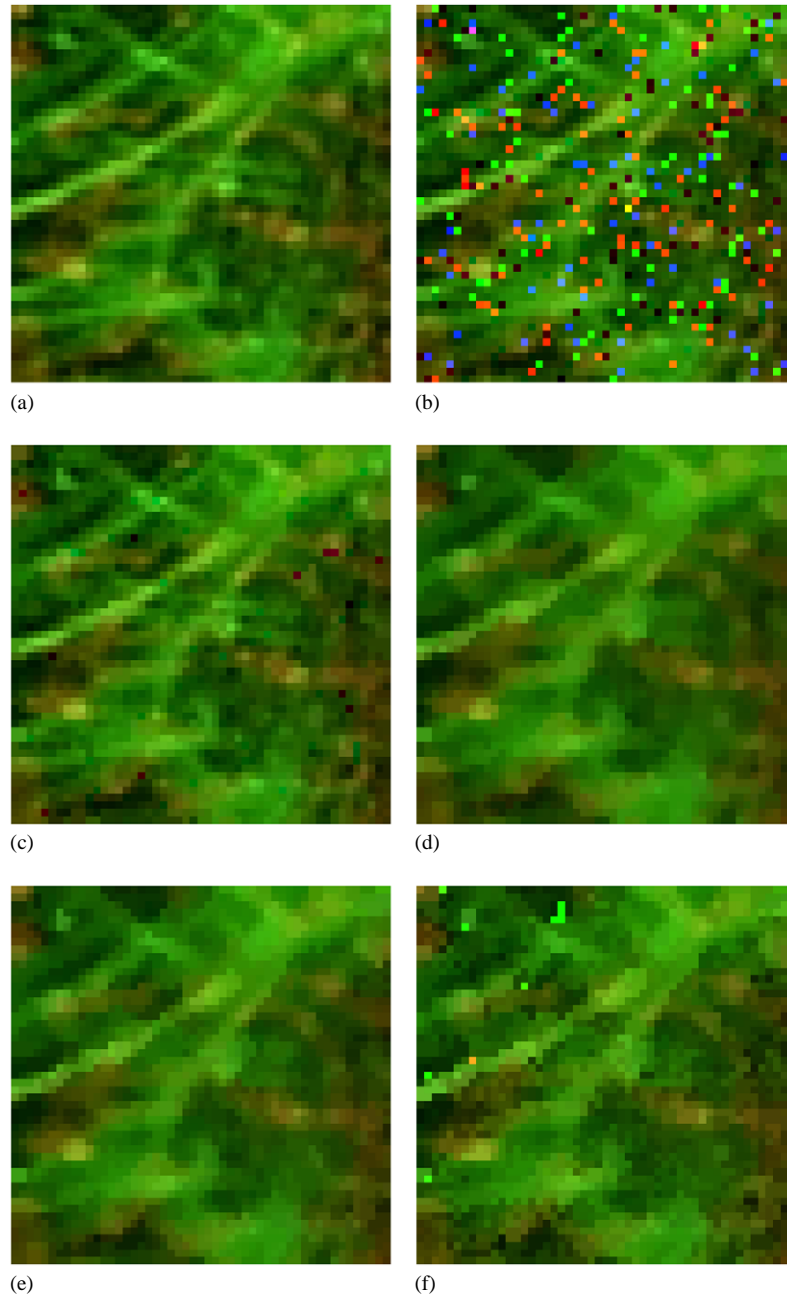
Fig. 16. Comparison of the new filter efficiency with the standard techniques: (a) test microscopic image, (b) image with 10% impulsive noise corruption (noise model II), (c) image after filtering with the proposed method, (d) VMF output, (e) DDF output, (f) BVDF output.

Although the algorithm may be of infinite length and may not converge to the optimal value of h it always provides a good approximation of h. To initiate the process, a starting interval $[a, b]$ and a predefined number of iterations should be provided by the designer. For a wide range of the fractions of noisy pixels (from $p = 0.01$ to more than 0.5) and majority of standard color images used for evaluation purposes $g(0)g(4) < 0$ holds, so a long enough interval is: $a = 0, b = 4$, (see Figs. 7 and 9).

Tables 3 and 4 show the results of the adaptive technique for three norms and LENA, PEPPERS

images with 11.5% pixels corrupted by 'salt and pepper' noise. It can be observed, that the optimal values of h are different for a specific image and vector norm.

In order to avoid the increase of computational complexity caused by the estimator, the following solution is recommended. For finding the optimal value of h using the method of bisection, not the whole image should be used but only a small part of it. For example, if an image is composed of 500×500 pixels, taking randomly placed 25×25 rectangle gives 625 pixels, which is enough for the purpose of estimation

Fig. 17. Illustration of the efficiency of the new algorithm of impulsive noise reduction in color images: (a) test aerial image, (b) images corrupted by 5% impulsive noise (model II), (c) new filter output, (d) effect of the median filtering (3×3 mask).

and finding the h value. On the other hand, it is only 0.25% of the pixels, so due to estimation and finding h (eight iterations), filtration time is extended only by about 2%.

6. Results

In order to evaluate the efficiency of the proposed filter, a number of simulations with different noise models presented in Section 3 were carried out. We compared the results obtained with the new filter with a set of standard noise reduction methods listed in Table 1.

The root of the mean squared error (*RMSE*), peak signal to noise ratio (*PSNR*) and normalized mean square error (*NMSE*) [24,3] were used for the comparisons. The simulation results shown in Tables 7 and 8 obtained using the noise model I(b) and II show that the new filter framework excels significantly over the standard techniques widely used in many multichannel image denoising applications.

As it is difficult to compare the results with all existing filters, the filtering efficiency can be judged from the direct comparison of results obtained using the new filter with the standard VMF, which can be treated as a reference filter. Through the VMF, the new method can be easily compared with other existing filtering algorithms, as the VMF is in many publications treated as a reliable benchmark.

The excellent results presented in Tables 2–4 are also confirmed in Figs. 10 and 11, where the new filter has been compared with VMF, BVDF and DDF using noise models Ia and II and the *PSNR* as a quality measure.

Another good property of the new adaptive filter is presented in Fig. 12, which shows that the new filter can be applied in an iterative way and that after the second or third iteration no further filtering is performed (the *PSNR* is not decreasing, as it is the case for the VMF, which indicates that the new filter reaches its root very quickly).

The good performance of the proposed adaptive filter design is also confirmed by subjective, visual comparison with the VMF presented in Figs. 13–15 using different noise corruption schemes. It can be easily observed, that the new filter has a good ability to distinguish between the corrupted and undisturbed pixel images, which is especially visible when evaluating the filters' estimation errors in Figs. 13 e and f and 14 e and f. As shown in Fig. 13 the new adaptive filter can be also successfully applied to gray-scale images.

As can be observed evaluating Figs. 16 and 17 presenting the efficiency of the proposed algorithm when applied to real life images, the new technique can be used in different applications, which are based on multichannel information.

Although the proposed filter has been applied for color images, its extension to images consisting of many channels is straightforward and preliminary results confirm, as expected, the excellent behavior of the proposed filtering framework.

7. Conclusions

The new algorithm presented in this paper is based on the concept of similarity between pixels, non-parametric estimation and the leave-one-out scheme, but can also be seen as a modification and improvement of the commonly used vector median filter (VMF). The computational complexity of the new filter is significantly lower than that of the VMF, especially when the four-neighborhood system is applied. The comparison shows that the new filter outperforms the VMF, as well as other standard procedures used in color image processing in terms of objective and subjective quality measures.

The algorithm is simple and fast and can be easily implemented. The proposed robust method of the estimation of noise intensity enables the tuning of the filter design parameter h to the image structure and noise statistics. The new filter can be applied in many applications, in which fast and reliable removal of impulses is required with minimal image quality degradation.

References

[1] Lukac R, Smolka B, Plataniotis KN, Venetsanopoulos AN. Color image filtering and enhancement (Special issue on color image processing). IEEE Signal Processing Magazine 2004;21: to appear.

[2] Pitas I, Venetsanopoulos AN. Nonlinear digital filters: principles and applications. Boston, MA: Kluwer Academic Publishers; 1990.

[3] Plataniotis KN, Venetsanopoulos AN. Color image processing and applications. Berlin: Springer; 2000.

[4] Karakos DG, Trahanias PE. Combining vector median and vector directional filters: the directional-distance filters. Proceedings of the IEEE Conference on Image Processing, ICIP-95, Oct. 1995. Washington, DC, p. 171–174.

[5] Lukac R, Smolka B, Plataniotis KN, Venetsanopoulos AN. Selection weighted vector directional filters (Special issue on colour for image indexing and retrieval). Computer Vision and Image Understanding 2003;to appear.

[6] Pitas I, Tsakalides P. Multivariate ordering in color image processing. IEEE Transactions on Circuits and Systems for Video Technology 1991;1(3):247–56.

[7] Pitas I, Venetsanopoulos AN. Order statistics in digital image processing. Proceedings of IEEE 1992;80(12):1893–923.

[8] Plataniotis KN, Androutsos D, Venetsanopoulos AN. Adaptive fuzzy systems for multichannel signal processing. Proceedings of the IEEE 1995;87(9):1601–22.

[9] Hardie RC, Arce GR. Ranking in R^P and its use in multivariate image estimation. IEEE Transactions on Circuits and Systems for Video Technology 1991;1(2):197–209.

[10] Lukac R. Adaptive vector median filtering. Pattern Recognition Letters 2003;24(212):1889–99.

[11] Lukac R. Adaptive color image filtering based on center-weighted vector directional filters. Multidimensional Systems and Signal Processing 2004;15:(to appear).

[12] Trahanias PE, Venetsanopoulos AN. Vector directional filters: a new class of multichannel image processing filters. IEEE Transactions on Image Processing 1993;2(4):528–34.

[13] Karakos D, Trahanias PE. Generalized multichannel image filtering structures. IEEE Transactions on Image Processing 1997;6(7):1038–45.

[14] Trahanias PE, Karakos D, Venetsanopoulos AN. Directional processing of color images: theory and experimental results. IEEE Transactions on Image Processing 1996;5(6):868–80.

[15] Vardavoulia MI, Andreadis I, Tsalides Ph. A new vector median filter for color image processing. Pattern Recognition Letters 2001;22:675–89.

[16] Smolka B, Chydzinski A, Wojciechowski K, Plataniotis K, Venetsanopoulos AN. On the reduction of impulsive noise in multichannel image processing. Optical Engineering 2001;40(6): 902–8.

[17] Smolka B, Plataniotis KN, Chydzinski A, Szczepanski M, Venetsanopoulos AN, Wojciechowski KW. Self-adaptive algo-

rithm of impulsive noise reduction in color images. Pattern Recognition 2002;(35):1771–84.

[18] Astola J, Haavisto P, Neuovo Y. Vector median filters. IEEE Proceedings 1999;78:678–89.

[19] Gabbouj M, Cheickh FA. Vector median—vector directional hybrid filter for colour image restoration. Proceedings of EUSIPCO 1996;96:879–81.

[20] Plataniotis KN, Androutsos D, Venetsanopoulos AN. Fuzzy adaptive filters for multichannel image processing. Signal Processing Journal 1996;55(1):93–106.

[21] Plataniotis KN, Androutsos D, Sri V, Venetsanopoulos AN. A nearest neighbour multichannel filter. Electronic Letters 1995; 31(22):1910–1.

[22] Scott DW. Multivariate density estimation. New York: Wiley; 1992.

[23] Silverman BW. Density estimation for statistics and data analysis. London: Chapman and Hall; 1986.

[24] Plataniotis KN, Androutsos D, Vinayagamoorthy S, Venetsanopoulos AN. Color image processing using adaptive multichannel filters. IEEE Transactions on Image Processing 1997;6(7): 933–50.