

Fast adaptive switching technique of impulsive noise removal in color images

Lukasz Malinski¹ · Bogdan Smolka²

Received: 29 September 2015 / Accepted: 22 April 2016 / Published online: 9 May 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract In the paper, a family of switching filters designed for the impulsive noise removal in color images is analyzed. The framework of the proposed denoising techniques is based on the concept of cumulated distances between the processed pixel and its neighbors. To increase the filtering efficiency, a robust scheme, in which the sum of distances to only the most similar pixels of the neighborhood serves as a measure of impulsiveness, was elaborated. As this trimmed measure is dependent on the image local structure, an adaptive mechanism was also incorporated. Additionally, a very fast design, which enables image denoising in practical applications, is proposed and the choice of the filter output, which is used to replace the noisy pixels, is discussed. The described family of filters was evaluated on a large set of natural test images and compared with the state-of-the-art restoration methods. The analysis of the achieved results shows that the novel filters outperform the existing techniques in terms of both denoising accuracy and computational complexity. In this

way, the proposed techniques can be recommended for the application in various image and video enhancement tasks.

Keywords Impulsive noise reduction · Color image enhancement and restoration · Image quality · Adaptive algorithm · Switching filter

1 Introduction

Noise reduction belongs to the most important image processing operations. The image restoration and enhancement methods are mainly relevant due to the miniaturization of high-resolution, low-cost image sensors, which frequently operate in poor lighting conditions.

Quite often, color images are corrupted by various types of noise, introduced by imperfections in sensors which influence the image formation process, signal instabilities, aging of the storage material, flawed memory locations, transmission errors in noisy channels and electromagnetic interferences. The quality of color images is severely decreased by impulsive noise distortions, and their removal is one of the most frequently performed low-level processing tasks [1–5].

The reduction in the disturbances introduced by the impulsive noise is crucial for the image preprocessing, as the corruption may have a significant negative impact on the success of the whole processing pipeline. Therefore, plentiful filtering techniques for impulsive noise suppression were developed during the many years of intensive research.

Numerous filters, which were designed to deal with impulsive noise in color images, are based on order statistics [6–11]. The majority of these algorithms relies on the ordering of a set of color pixels, treated as vectors,

Electronic supplementary material The online version of this article (doi:10.1007/s11554-016-0599-6) contains supplementary material, which is available to authorized users.

This work was supported by the Polish National Science Center (NCN) under the Grant: DEC-2012/05/B/ST6/03428 and POIG.02.03.01-24-099/13 grant: GeCONiI—Upper Silesian Center for Computational Science and Engineering.

✉ Lukasz Malinski
lukasz.malinski@polsl.pl

¹ Division of Industrial Informatics, The Silesian University of Technology, Krasińskiego 8, 40-019 Katowice, Poland

² Institute of Automatic Control, The Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

belonging to the processed pixel's neighborhood, represented by sliding operational window W . For each pixel from W , the sum of distances to other samples belonging to the same window is assigned and then the cumulated distances are sorted. As a result, an ordered sequence of color pixels is obtained, which is the basis for various filtering methods.

One of the most popular methods based on reduced ordering, used in many filtering designs, is the Vector Median Filter (VMF) [6, 12]. The VMF output is the pixel from W for which the sum of cumulated distances to other samples is minimized. It is always one of the pixels of the filtering window, which is profitable as the filter does not introduce any new colors to the processed image. However, when all pixels of W are affected, for example by additional Gaussian noise, the output is also noisy. Numerous solutions devoted to the elimination of this undesired behavior were introduced, resulting in significantly better filtering performance [12–15]. To increase the VMF efficiency, weights are assigned to the distances between pixels, which privilege the central pixel of the filtering window, thus diminishing the number of unnecessarily altered pixels [16, 17].

The efficiency of the techniques utilizing various vector ordering schemes is limited due to a common feature—every pixel of the image is processed, regardless whether it is contaminated or not. This results in the inevitable distortion of uncorrupted pixels and degradation of image quality. Therefore, a natural improvement has been made introducing more efficient *switching filters* [18–23], which aim at the restoration of only the polluted pixels, leaving the uncorrupted ones unaltered. In the majority of the switching techniques, there is a need to determine the dissimilarity between the color pixels. The most intuitive and popular approach is to compute the Euclidean distance in the RGB color space; however, there are many other measures of vector dissimilarity applied in various filtering frameworks [24–26].

Further improvement resulting in better robustness to the occurrence of outliers was achieved by calculation of only a few smallest distances between a pixel and other samples belonging to the same processing window [27–29]. Such modification, in which the trimmed cumulative distance is utilized as a measure of pixel corruption, also facilitates the preservation of the original image edges and tiny details.

The decision-making step, differentiating between the distorted and uncorrupted pixels, seems to be more important than the choice of the algorithm used for the replacement of pixels classified as noise. The reason is simple—more precise impulse detection process results in less unwanted original pixels alteration.

There are numerous noisy pixel detection schemes proposed in the literature, and among switching filters, several groups of filtering designs can be enumerated. The Sigma Vector Median Filter (SVMF) [30, 31] and Adaptive Vector Median Filter (AVMF) [18] can be regarded as popular representatives of techniques based on reduced ordering statistics.

An efficient family of filters based on the *peer group* framework was proposed in [32]. The idea of this switching strategy can be found in various works [33–35]. Also significant improvement has been made introducing the Fast Peer Group Filter (FPGF) [34]. It was also an inspiration of the recently proposed Fast Averaging Peer Group Filter (FAPGF) [36], which delivers a very good performance for highly contaminated images.

Another group of switching filters dedicated to the suppression of the impulsive noise in color images is based on the elements of the quaternion theory [37, 20, 38]. The color pixels, which are generally represented by three channels in the RGB color space, are expressed as quaternions without the real component. In this way, the similarity between pixels is defined in the quaternion form and is used as an alternative for the Euclidean distance, commonly used in the popular filtering designs.

The methods based on fuzzy set theory were also elaborated for the impulsive noise removal [39–45]. These algorithms proved to be very flexible and offer a powerful performance not only for single image applications, but also for the enhancement of video sequences.

The filters proposed in this paper belong to the family of switching techniques. The impulse detection step is based on the reduced ordering and computation of trimmed cumulative Euclidean distances. Both Arithmetic Mean Filter (AMF) and VMF will be considered as the filter providing the estimate of the corrupted pixels, to enable a comparison of these two competitive solutions.

2 Adaptive switching filtering design

Most of the filtering techniques determine their output for the pixel located at position (u, v) using n samples belonging to a sliding, operating window W with $\mathbf{x}_{u,v}$ at its center. In order to simplify further analysis, the pixels belonging to W will be denoted as $\mathbf{x}_1, \dots, \mathbf{x}_n$, and \mathbf{x}_1 will be the central pixel of W as shown in Fig. 1.

The reduced ordering scheme operates on the sum of dissimilarity measures (distances) denoted as d , between a given pixel and the samples from the filtering window W . In this way, the cumulated dissimilarity measure D assigned to pixel \mathbf{x}_i , ($i = 1, \dots, n$) from W , is

x_2	x_3	x_4
x_5	$x_{u,v}$ x_1	x_6
x_7	x_8	x_9

Fig. 1 Notation of the pixels in the filtering window

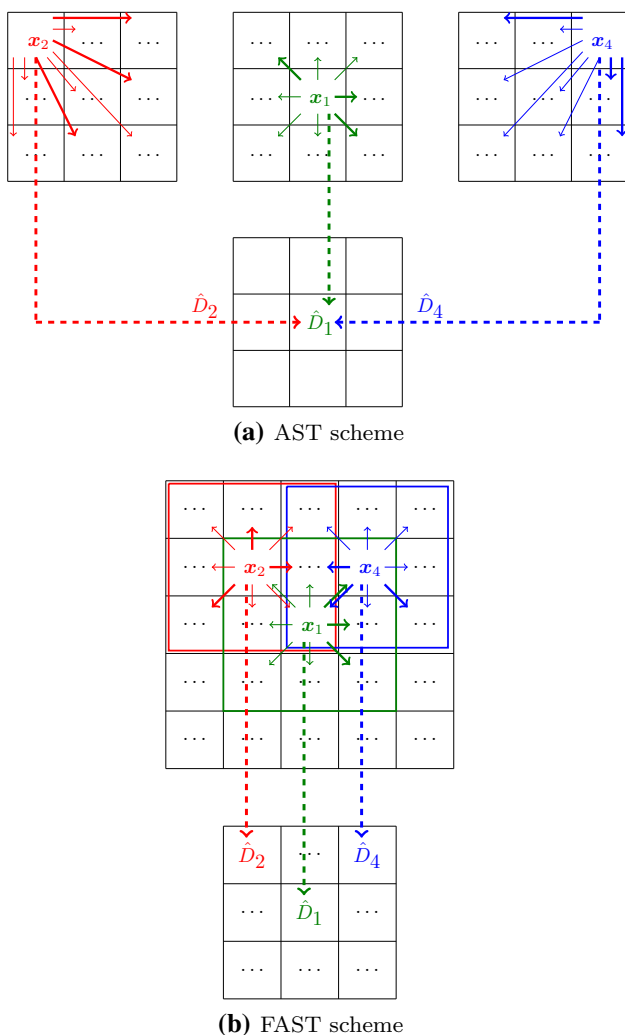


Fig. 2 Construction of the AST and FAST denoising schemes. The AST design (a) requires the calculation of a trimmed sum of distances for every pixel of W to determine their minimum value, which is assigned to the central pixel. The FAST scheme (b) needs only the distance values directly designated to image pixels and the minimum value is calculated in the local neighborhood of the processed sample

$$D_i = \sum_{j=1}^n d(x_i, x_j). \tag{1}$$

The distances $d_{ij} = d(x_i, x_j)$ between x_i and all other pixels x_j belonging to W , ($i, j = 1..n, i \neq j$) can be sorted in ascending order:

$$d_{i1}, \dots, d_{iv} \longrightarrow d_{i(1)}, \dots, d_{i(v)}, \tag{2}$$

where $v = n - 1$, and instead of the aggregated distances in (1) a trimmed sum of distances \hat{D} can be used [28]

$$\hat{D}_i = \sum_{r=1}^m d_{i(r)}, \tag{3}$$

where m denotes the number of nearest pixels taken for the calculation of the trimmed sum of distances and $d_{i(r)}$ is the r -the smallest dissimilarity measure. The trimmed sum \hat{D}_i is significantly less susceptible to outliers among pixels of W than the standard sum of distances D_i [46, 47].

The value of \hat{D}_1 , which is assigned to the central pixel x_1 can be treated as a measure of the pixel corruption. This value is low when there exist at least m similar pixels in the neighborhood, otherwise the central pixel x_1 may be considered as corrupted. If \hat{D}_1 divided by m is greater than a predefined threshold value T

$$\frac{\hat{D}_1}{m} > T, \tag{4}$$

then the central pixel of W will be considered noisy and will be replaced by the output of a suitable robust filter, otherwise this pixel will be designated as uncorrupted and remains unaltered. The division by m in (4) makes the T value independent on the number of close pixels

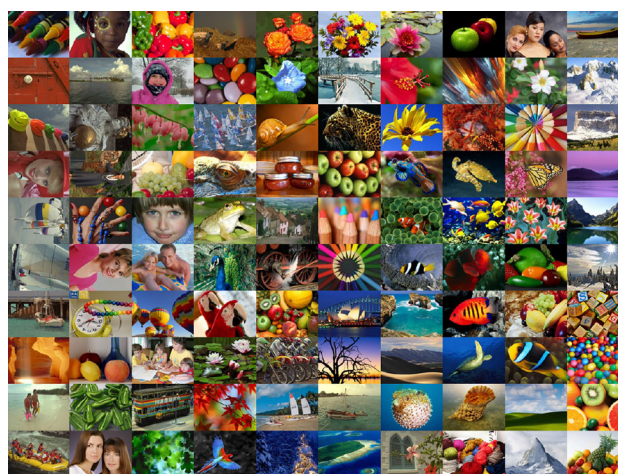


Fig. 3 Image database used for the analysis of parameter selection

taken for the calculation of \hat{D} . The described above decision-making scheme will be denoted as *Switching Trimmed* (ST).

Additionally, for every processed pixel the *map of noise* array M is updated

$$M_{u,v} = \begin{cases} 0 & : \text{ if recognised as corrupted,} \\ 1 & : \text{ otherwise.} \end{cases} \quad (5)$$

This map will be later used for the noisy pixels replacement.

Now, in order to address the effect of high values of the trimmed sum of distances \hat{D} in textured regions, a kind of adaptiveness can be introduced to the ST scheme by subtracting the minimum value $\hat{D}_{(1)}$ calculated for the pixels of W [27]. In this way, the modified condition (4) for the noisy pixel detection is

$$\frac{\hat{D}_1 - \hat{D}_{(1)}}{m} > T, \quad \hat{D}_{(1)} = \min\{D_i : i = 1, \dots, v\}. \quad (6)$$

This decision-making step becomes more robust and accurate in dealing with local image textural features and tiny details. The proposed scheme will be denoted as *Adaptive Switching Trimmed* (AST).

In the AST scheme, the trimmed sums have to be computed for every pixel in W in order to determine the minimum one. Despite the computational efficiency of this solution, which will be shown later, it requires a relatively large number of distance computations for every pixel in the processing window and additionally the minimum value has to be determined. Therefore, a simplified and faster approach has also been taken under consideration.

The fast AST (FAST) scheme can be performed in two steps. In the first step, for every image pixel, the computation of the trimmed sum of distances to its neighbors is performed: $\hat{D}_1 = \sum_{r=1}^m d(1, (r))$ [48].

In the second step, the minimum trimmed sum $\hat{D}_{(1)}$ is computed from all the values of \hat{D} assigned to the pixels belonging to W and the decision concerning the central pixel corruption is performed according to (6).

The AST and FAST schemes are summarized in Fig. 2. In the AST scheme (a), the trimmed sum of distances have to be computed for each pixel of the filtering window W and then the minimum value is calculated. The FAST scheme (b) requires only the calculation of the trimmed cumulative distances for each central pixel of W (each image pixel), and the minimum value of \hat{D} is taken from the values previously assigned to the pixels of W . Therefore, the FAST scheme is n times faster than AST, as only one trimmed distance measure has to be calculated for each pixel, instead of n values required in the AST scheme.



(a)



(b)

Fig. 4 Images used for the analysis of parameter selection. **a** MOTO-CROSS. **b** RAFTING

Finally, pixels labeled as corrupted ($M_{u,v} = 0$) are processed by one of the two algorithms, which are used to determine the estimate of the noisy pixel:

- VMF output replaces the central pixel of W with a pixel corresponding to $\hat{D}_{(1)}$,
- AMF output replaces the central pixel of W with the average of the pixels from W classified as not corrupted ($M_{u,v} = 1$). However, in rare situations, all pixels of the W can be detected as corrupted and in such situations the VMF output is used.

The resulting filters will be denoted as follows:

- STVMF—Switching Trimmed with VMF output,
- ASTVMF—Adaptive Switching Trimmed with VMF output,
- FASTVMF—Fast Adaptive Switching Trimmed with VMF output,
- STAMF—Switching Trimmed with AMF output,
- ASTAMF—Adaptive Switching Trimmed with AMF output,

Fig. 5 Dependence of PSNR quality measure on the parameters m and T obtained when applying the analyzed techniques to the test image RAFTING contaminated with impulsive noise of intensity $p = 0.2$. **a** STVMF. **b** STAMF. **c** ASTVMF. **d** ASTAMF. **e** FASTVMF. **f** FASTAMF

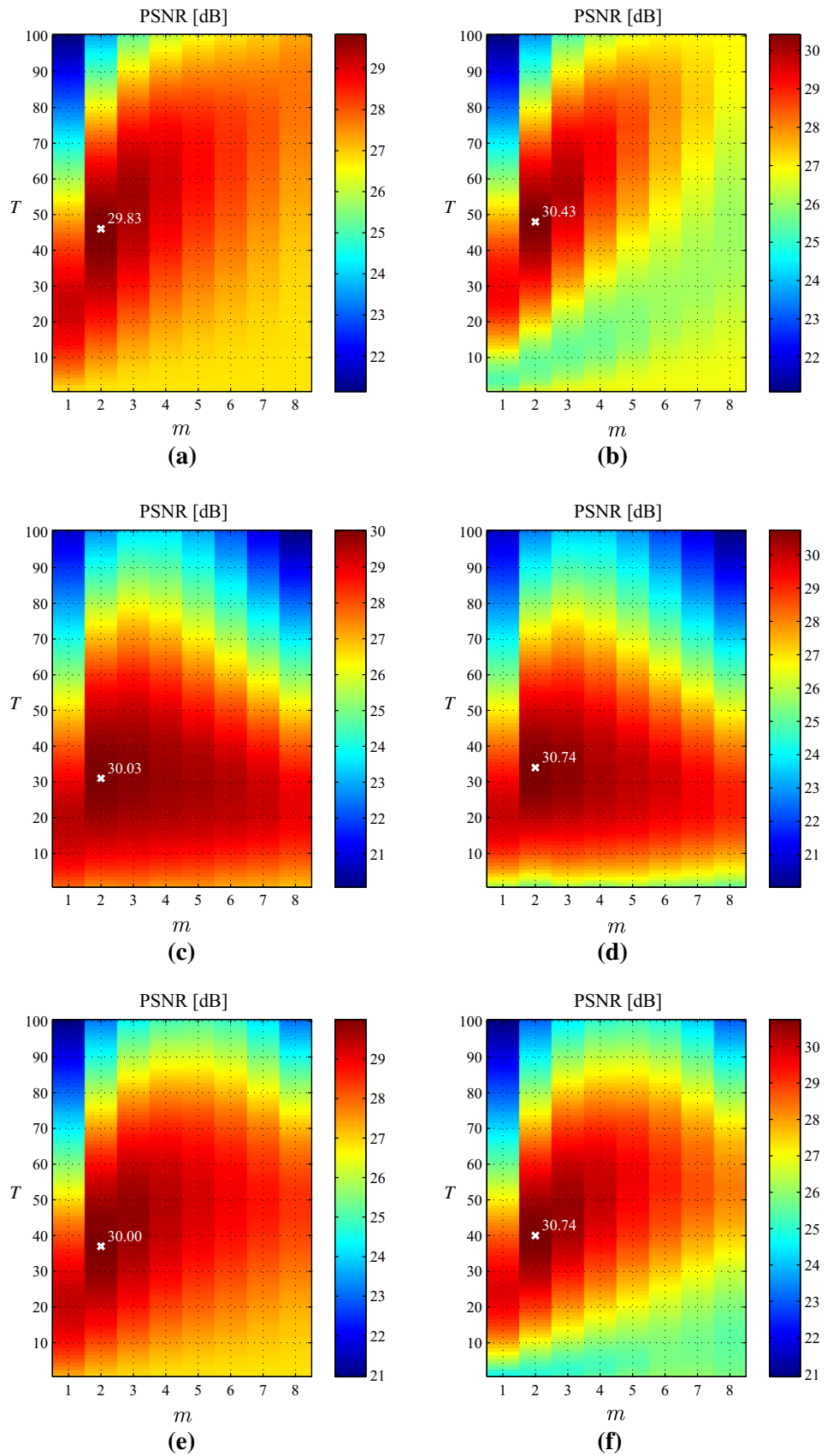


Fig. 6 Dependence of PSNR quality measure on the parameters m and T obtained when applying the analyzed techniques to the test image MOTOCROSS contaminated with impulsive noise of intensity $p = 0.2$. **a** STVMF. **b** STAMF. **c** ASTVMF. **d** ASTAMF. **e** FASTVMF. **f** FASTAMF

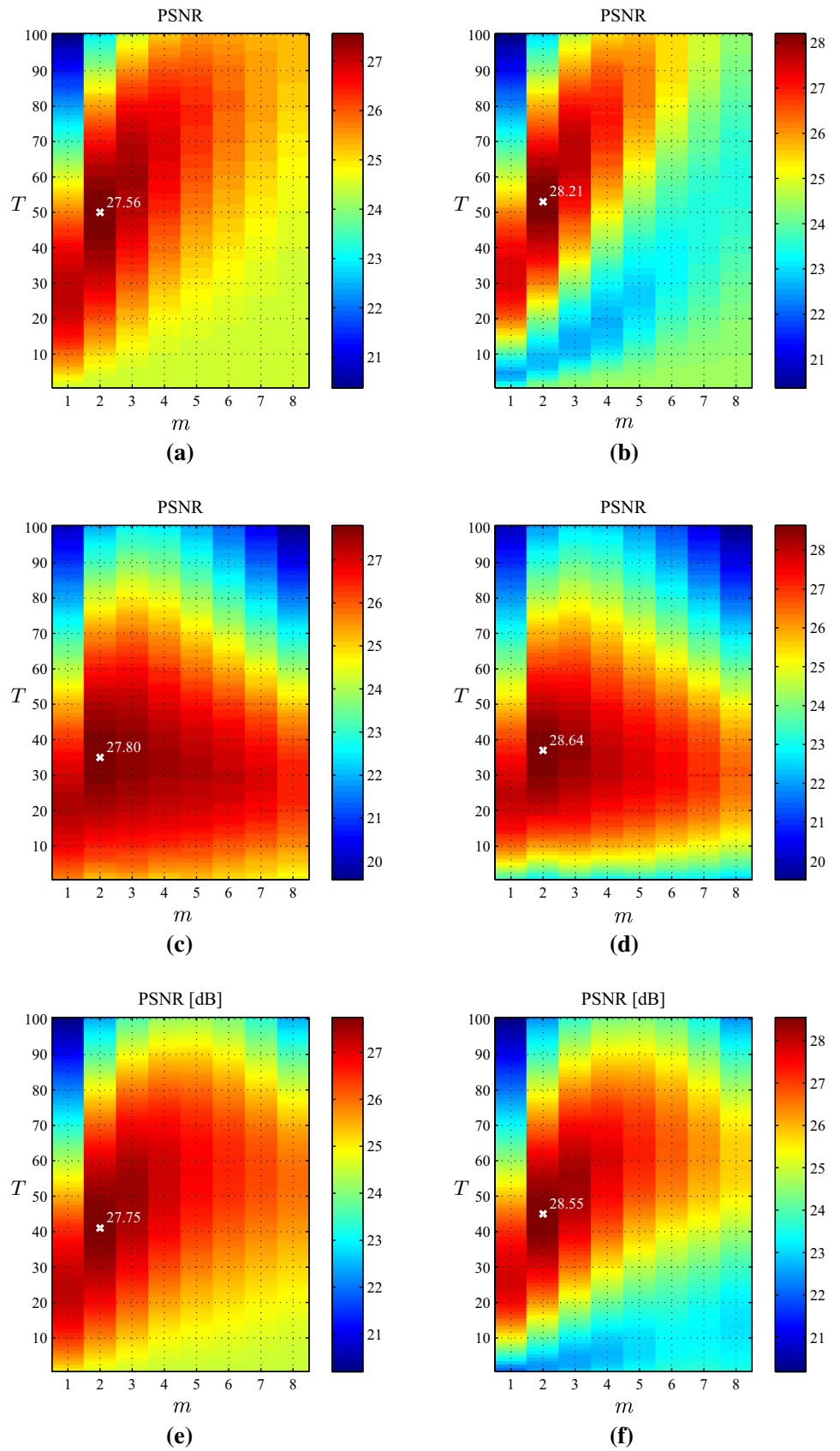
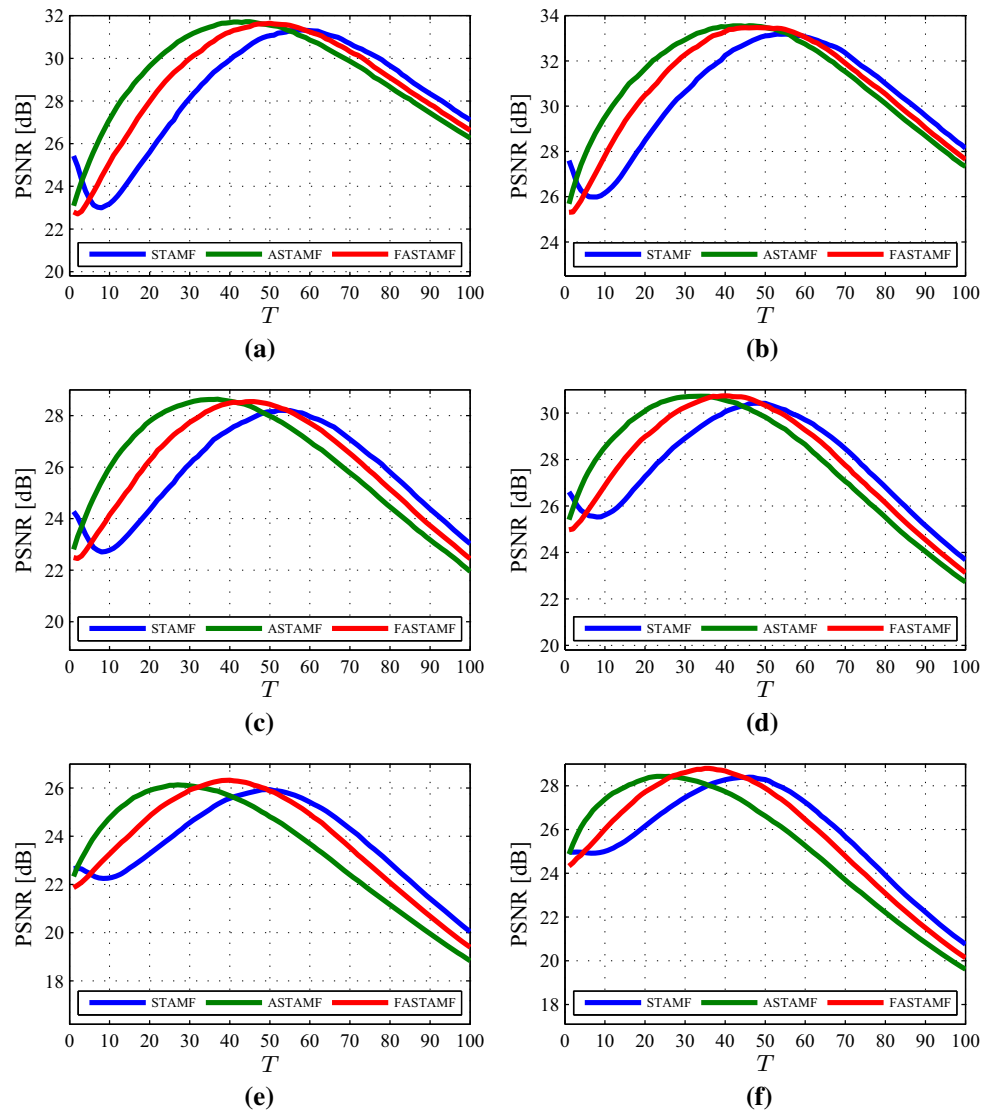


Table 1 Comparison of the efficiency of the pixel replacement techniques based on the VMF and AMF using the PSNR quality measure

p	STAMF	ASTAMF	FASTAMF	STVMF	ASTVMF	FASTVMF
MOTOCROSS						
0.1	31.33	31.72	31.64	31.06	31.33	31.26
0.2	28.21	28.64	28.55	27.56	27.80	27.75
0.3	25.94	26.13	26.33	24.58	24.72	24.71
RAFTING						
0.1	33.19	33.56	33.47	33.15	33.45	33.37
0.2	30.43	30.74	30.74	29.83	30.03	30.00
0.3	28.40	28.44	28.80	26.81	26.94	26.95

Bold values represent the results obtained for our algorithms

Fig. 7 Dependence of PSNR on the parameter T obtained for test images MOTOCROSS and RAFTING contaminated with noise intensity $p = 0.1, 0.2, 0.3$. **a** MOTOCROSS, $p = 0.1$ **b** RAFTING, $p = 0.1$ **c** MOTOCROSS, $p = 0.2$ **d** RAFTING, $p = 0.2$ **e** MOTOCROSS, $p = 0.3$ **f** RAFTING, $p = 0.3$



– FASTAMF—Fast Adaptive Switching Trimmed with AMF output.

There are many approaches to impulsive noise modeling [49, 50]. One of the most popular contamination models is

the so called color salt & pepper noise [51–53], which assumes that a fraction of the image pixels denoted as p is corrupted in such a way that the RGB channels are assigned either the minimum or maximum value of the

allowable dynamic range, (0 or 255 assuming 8 bit channel representation). The pixel modified by the impulsive noise can be totally corrupted, so that all three channels are replaced by the extreme values, but also one or two components of a corrupted pixel can remain unchanged. The noise distortion can be fully correlated (three channels are always affected), or it can be modeled according to a predefined correlation of channel contamination.

The removal of the salt & pepper noise is facilitated by the fact that only the extreme values of the corrupted pixels should be restored. Therefore, a more challenging and realistic noise model assumes that all channels of an affected pixel are replaced by a random variable drawn from the uniform distribution. For the experiments reported in this paper, we assume that the affected pixels have the RGB channels changed independently by values from the range (0, 255) [36, 49, 54]. This corruption scheme will be called *uniform noise model—UNM*.

3 Parameter selection

In order to determine the recommended values of the parameters m and T of the described filtering techniques, a large number of simulations were performed on the image database consisting of 100 true color test images of size 640×480 depicted in Fig. 3. This set of images is made available as *Electronic Supplementary Material* and is accompanied by a file which provides for each image its entropy measure and the number of unique colors.

The test images were contaminated with UNM impulsive noise of 3 different intensity levels: $p = \{0.1, 0.2, 0.3\}$. For the evaluation of the image restoration performance, following quality metrics were used:

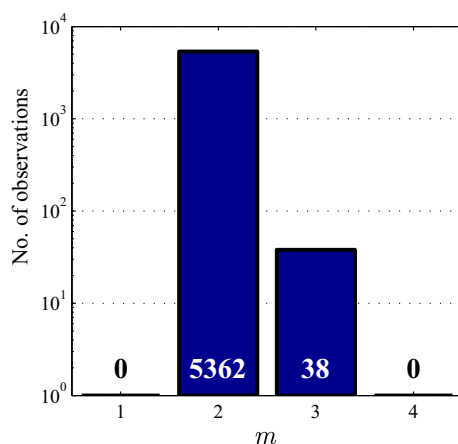


Fig. 8 Histogram of the optimal m parameter values obtained for all 5400 (3 measures \times 3 contamination levels \times 6 algorithms \times 100 images) tested cases

– Peak Signal-to-Noise Ratio (PSNR)

$$\text{MSE} = \frac{1}{3N} \sum_{j=1}^N \sum_{q=1}^3 (x_{j,q} - \hat{x}_{j,q})^2, \quad \text{RMSE} = \sqrt{\text{MSE}}, \quad (7)$$

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right) = 20 \log_{10} \frac{255}{\text{RMSE}},$$

where $x_{j,q}$, $q = 1, 2, 3$, are the channels of the original image pixels indexed by j , N is the number of image pixels and $\hat{x}_{j,q}$ are the restored components.

– Mean Average Error (MAE):

$$\text{MAE} = \frac{1}{3N} \sum_{j=1}^N \sum_{q=1}^3 |x_{j,q} - \hat{x}_{j,q}|, \quad (8)$$

– Normalized Color Distance (NCD) [1]:

$$\text{NCD} = \frac{\sum_{j=1}^N \|\mathbf{x}_{\text{Lab}} - \hat{\mathbf{x}}_{\text{Lab}}\|}{\sum_{j=1}^N \|\mathbf{x}_{\text{Lab}}\|}, \quad (9)$$

where \mathbf{x}_{Lab} and $\hat{\mathbf{x}}_{\text{Lab}}$ are the components of the original and restored image pixels in the CIE Lab color space and $\|\cdot\|$ denotes the Euclidean norm.

Due to a large number of the results obtained using different quality metrics and their very similar qualitative characteristics, only the PSNR measure will be used in the analysis presented in this Section.

The images corrupted according to the UNM with 3 different intensity levels were processed with all 6 described above filtering techniques. Each image was contaminated 10 times with different seeds of the random number generator and the obtained quality measures were then averaged. The denoising process was performed for each pair of parameters $\{m, T\}$, where $m \in \langle 1, 8 \rangle$ and $T \in \langle 1, 100 \rangle$.

The first (more detailed) step of the filters' analysis focuses on 2 exemplary, natural test images: RAFTING and MOTOCROSS of size 640×480 shown in Fig. 4. For these images, the diagrams showing filtering performance for each tested pair of parameters and contamination intensity $p = 0.2$ are presented in Fig. 5 (RAFTING) and in Fig. 6 (MOTOCROSS).

The visual analysis implies following remarks:

- The filters with AMF output have better peak performance for an optimal pair of parameters. Another argument in favor of such a statement is presented in Table 1, where optimal values of PSNR obtained for both images, and all algorithms and all contamination levels are presented.
- The filters using VMF output possess better tolerance for the selection of parameters (choosing other than

Fig. 9 Box-plots of the optimal T values determined using PSNR quality measure for all 100 test images, **a** $p = 0.1$, **b** $p = 0.2$, **c** $p = 0.3$

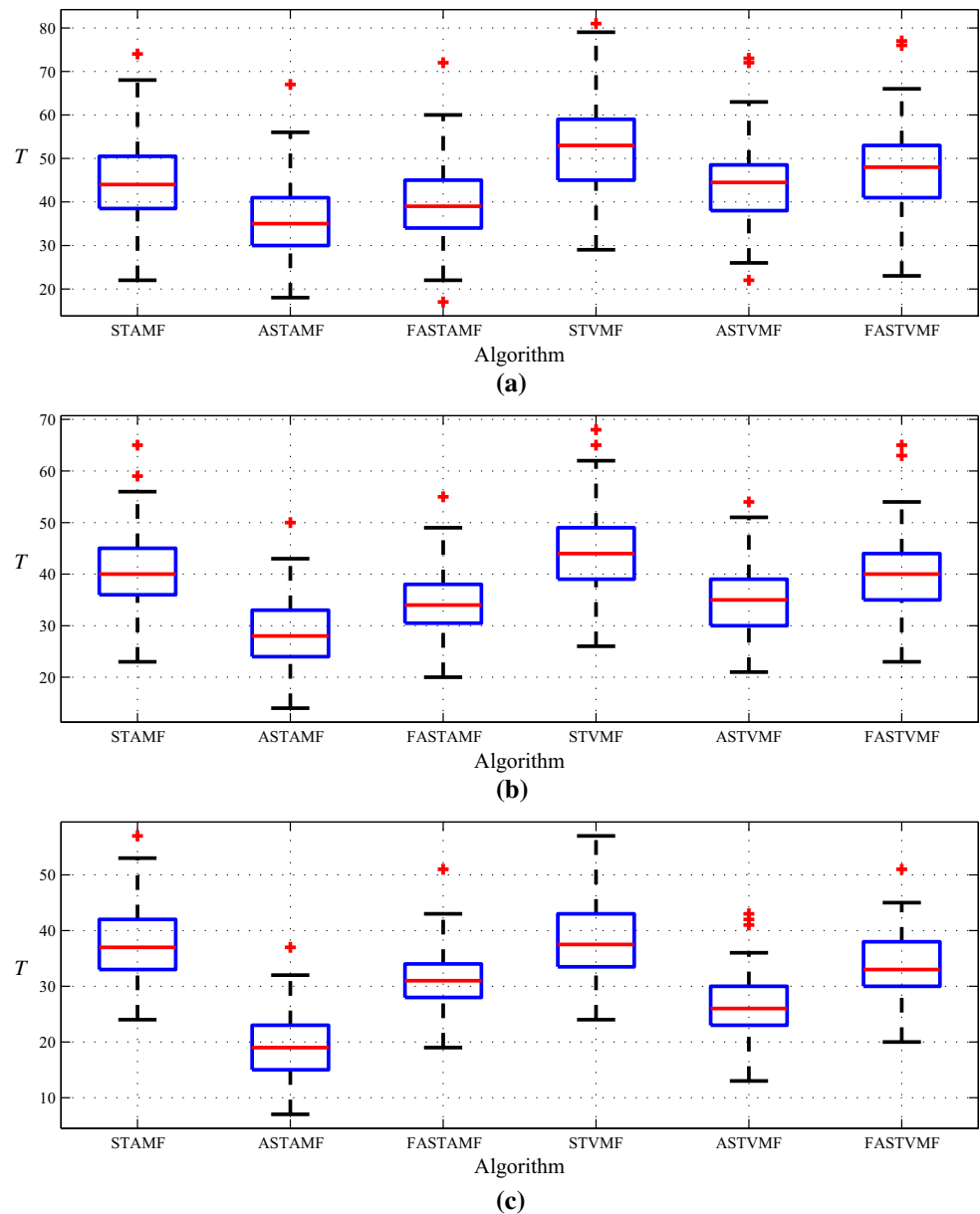


Table 2 Medians with interquartile ranges of optimal T values maximizing the PSNR index

p	STAMF	ASTAMF	FASTAMF	STVMF	ASTVMF	FASTVMF
0.1	39.0 (11.0)	44.0 (12.0)	35.0 (11.0)	48.0 (12.0)	53.0 (14.0)	44.5 (11.5)
0.2	34.0 (7.5)	40.0 (9.0)	28.0 (9.0)	40.0 (9.0)	44.0 (10.0)	35.0 (9.0)
0.3	31.0 (6.0)	37.0 (9.0)	19.0 (8.0)	33.0 (8.0)	37.0 (9.5)	26.0 (7.0)

optimal values result in lower loss of filtering efficiency).

- The value of the m parameter, optimizing the PSNR measure, in each case was equal to 2.

As filters with AMF output achieve better peak performance, the more detailed analysis of the impact of T parameter on their efficiency for STAMF, ASTAMF

and FASTAMF techniques is exhibited in Fig. 7. The presented plots enable to draw the following conclusions:

- The PSNR as a function of T is smooth and slowly varying. Therefore, a deviation from optimal T parameter setting does not result in significant loss of filtering performance.

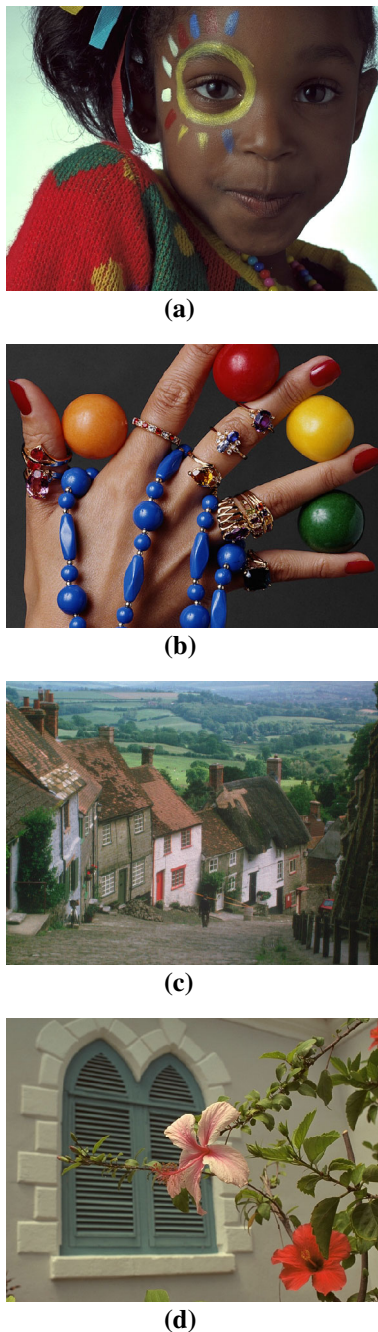


Fig. 10 Color test images used for comparison with the state-of-the-art filters. **a** GIRL. **b** HAND. **c** GOLDHILL. **d** FLOWER

- The optimal T values for various algorithms are clearly different. The ASTAMF and FASTAMF schemes require a lower value of threshold T than STAMF, which is caused by subtraction of the minimal trimmed accumulated distance measure in (6).
- It is difficult to determine which filter achieves best efficiency, as their relative performance is different for various contamination levels. The AST scheme yields better results for low noise intensity,

while FAST scheme takes a lead for stronger image corruption.

- Although the ST scheme may achieve best overall performance for very low and high T values (deviating much from those recommended), there is no evidence that its peak performance may outperform the AST and FAST techniques for low noise intensities ($p < 0.3$).

The second (more global) step of our analysis considers all of the tested images. The optimal values of parameters $\{m, T\}$, maximizing the PSNR measure, were determined for each of the analyzed filters and noise contamination level.

The histogram of optimal values of m is presented in Fig. 8. It is clear that $m = 2$ is a value to be recommended for all filters and all contamination intensities. The statistically insignificant occurrence of $m = 3$ implies that more detailed analysis of this parameter has no relevance.

On the other hand, the proper recommendation of the T values is not so unequivocal. Figure 9 depicts the box-plots presenting medians and quartiles for all optimal T obtained with regard to different contamination levels and algorithms. The numeric values of medians and interquartile ranges (IQR) are gathered in Table 2. We observed that bigger values of the threshold T are usually needed for highly textured images. This indicates that the local image entropy could be incorporated into the algorithm for adaptive, structure dependent tuning of the thresholding parameter.

The medians of T are slightly declining with the increase in noise intensity. Therefore, the medians achieved for $p = 0.2$, which can be considered as a medium corruption, should be considered as recommended values of the T parameter. In this way, the suggested values of T are: 34 for STAMF, 40 for STAMF and 28 for FASTAMF.

4 Comparison with the state-of-the-art denoising methods

An indispensable final step in the development of any new filtering technique is the comparison with other competitive solutions available in the rich literature. Among filters proposed in this paper, only those with AMF output have been taken for comparison due to their higher efficiency. The described filtering designs were compared with the following methods, which are known to deliver very satisfying denoising performance:

- Adaptive Central-Weighed VMF (ACWVMF) [55],
- Fast Averaging Peer Group Filter (FAPGF) [36],
- Fast Fuzzy Noise Reduction Filter (FFNRF) [56],
- Fuzzy Ordered Vector Median Filter (FOVMF) [57],
- Fast Peer Group Filter (FPGF) [34],

Table 3 Comparison of the efficiency of the analyzed algorithms with the state-of-the-art techniques for the GIRL color test image

Measure	p	New algorithms			State-of-the-art algorithms					
		STAMF	ASTAMF	FASTAMF	ACWVMF	FAPGF	FFNRF	FOVMF	FPGF	SVMFr
PSNR	0.1	37.43	37.78	37.68	35.62	36.77	35.85	32.18	35.54	33.89
	0.2	34.81	34.97	35.13	29.51	34.34	32.56	30.65	31.75	27.28
	0.3	32.25	31.44	32.66	23.48	32.06	27.65	26.76	26.31	21.31
	0.4	29.56	26.36	29.49	19.17	29.57	22.60	22.22	21.39	17.18
	0.5	25.36	21.28	24.45	15.81	25.94	18.32	18.24	17.41	14.06
NCD (E-04)	0.1	44.80	42.47	43.32	54.53	49.12	52.57	283.83	56.12	83.53
	0.2	89.50	86.01	86.40	149.62	100.39	112.65	318.72	122.62	199.97
	0.3	142.88	150.44	137.63	380.84	168.82	229.84	423.16	265.83	533.75
	0.4	215.56	311.38	218.46	824.93	271.63	491.39	690.49	590.24	1163.36
	0.5	363.56	706.32	436.26	1596.82	466.78	1028.93	1248.50	1234.76	2170.63
MAE	0.1	0.54	0.52	0.52	0.64	0.59	0.64	3.20	0.73	1.02
	0.2	1.03	1.00	0.99	1.60	1.13	1.20	3.61	1.48	2.07
	0.3	1.61	1.66	1.54	3.99	1.79	2.30	4.79	3.11	5.33
	0.4	2.38	3.17	2.36	8.70	2.74	4.85	7.77	6.75	11.63
	0.5	4.00	7.03	4.48	16.91	4.55	10.27	14.13	14.04	21.83
FSIMc	0.1	0.9956	0.9959	0.9959	0.9763	0.9950	0.9775	0.9689	0.9768	0.9729
	0.2	0.9911	0.9913	0.9919	0.9519	0.9902	0.9703	0.9593	0.9644	0.9366
	0.3	0.9825	0.9772	0.9841	0.8734	0.9799	0.9421	0.9168	0.9140	0.8336
	0.4	0.9667	0.9243	0.9651	0.7554	0.9583	0.8610	0.8185	0.8055	0.7018
	0.5	0.9181	0.8077	0.8970	0.6296	0.9050	0.7307	0.6859	0.6668	0.5796

Bold values represent the best results obtained (in a corresponding row)

– Ranked Sigma Vector Median Filter (SVMFr) [30].

In addition to quality measures presented in the previous Section, the Feature Similarity index (FSIMc) [58, 59] was also used to compare the efficiency of the evaluated filters. The structural similarity measures, like FSIMc, are highly correlated with the human visual system, which make them very useful for the analysis of noise suppression efficiency [60, 61].

The efficiency tests were performed using four selected images corrupted 10 times with UNM noise of intensities $p = 0.1, 0.2, \dots, 0.5$, (Fig. 10). Every contaminated image was denoised, and the outcome of each method was evaluated using averaged PSNR, NCD, MAE and FSIMc dissimilarity measures. The results obtained for each image are summarized in Tables 3, 4, 5 and 6.

The visual comparison of the performance of the analyzed filters is presented for image GIRL (Fig. 11) and HAND (Fig. 12). The comparison of the efficiency of the denoising methods is presented in terms of PSNR measure for all four test images in Fig. 13.

Finally, the obtained results can be summarized as follows:

- For low contamination levels ($p < 0.3$), ASTAMF algorithm clearly excels other techniques. However, the fast version of this algorithm (FASTAMF) is not far behind.
- For medium noise intensity ($p = 0.3$), the FASTAMF algorithm takes a lead for all tested images and this observation is valid for all computed dissimilarity measures.
- In case of images corrupted by stronger noises ($p > 0.3$), the FASTAMF competes with STAMF algorithm, which becomes surprisingly more efficient than others. This observation suggests that the adaptation mechanism becomes less important or even undesirable for more extreme noise occurrence.
- If only PSNR measure is considered, the FAPGF algorithm shows a competitive performance for medium noise levels.
- For very high noise corruption ($p = 0.5$), the adaptive designs of ASTAMF and FASTAMF loose their power and the STAMF yields better results.
- Generally, the thresholding parameter is only slightly dependent on the structure of natural images. Higher

Table 4 Comparison of the efficiency of the analyzed algorithms with the state-of-the-art techniques for the HAND color test image

Measure	p	New algorithms			State-of-the-art algorithms					
		STAMF	ASTAMF	FASTAMF	ACWVMF	FAPGF	FFNRF	FOVMF	FPGF	SVMFr
PSNR	0.1	31.17	33.07	32.64	32.19	30.94	31.57	29.33	30.88	31.74
	0.2	29.61	31.05	30.77	28.08	29.59	29.42	28.13	28.71	26.61
	0.3	28.11	28.77	29.24	23.45	28.39	26.21	25.77	25.42	21.48
	0.4	26.31	25.18	26.82	19.51	26.63	22.25	22.30	21.47	17.58
	0.5	23.96	21.34	23.67	16.40	24.46	18.62	18.90	18.01	14.61
NCD (E-04)	0.1	93.98	76.17	79.79	88.97	102.45	90.45	423.48	118.92	111.50
	0.2	164.64	143.55	147.17	211.53	178.81	177.22	474.88	215.58	256.01
	0.3	244.62	233.37	221.62	457.77	270.25	321.33	588.94	382.86	605.07
	0.4	354.89	426.63	342.24	935.63	413.87	614.34	868.06	737.65	1260.56
	0.5	530.60	829.24	576.06	1689.18	632.33	1151.32	1399.39	1369.14	2242.48
MAE	0.1	1.01	0.78	0.83	0.89	1.11	0.91	3.69	1.26	1.08
	0.2	1.63	1.36	1.42	1.93	1.75	1.60	4.19	2.12	2.25
	0.3	2.33	2.13	2.07	4.16	2.49	2.80	5.26	3.66	5.32
	0.4	3.29	3.74	3.08	8.49	3.61	5.31	7.85	6.97	11.17
	0.5	4.84	7.13	5.06	15.56	5.34	10.08	12.99	13.02	20.22
FSIMc	0.1	0.9932	0.9952	0.9948	0.9855	0.9926	0.9856	0.9776	0.9845	0.9836
	0.2	0.9885	0.9911	0.9909	0.9696	0.9882	0.9800	0.9705	0.9758	0.9606
	0.3	0.9821	0.9809	0.9849	0.9200	0.9809	0.9613	0.9454	0.9447	0.8913
	0.4	0.9672	0.9422	0.9674	0.8318	0.9625	0.9052	0.8801	0.8703	0.7858
	0.5	0.9376	0.8614	0.9221	0.7281	0.9290	0.8091	0.7824	0.7645	0.6763

Bold values represent the best results obtained (in a corresponding row)

values of T may yield better results for images with high frequency texture or containing tiny details.

5 Computational complexity

Although the noise suppression efficiency, expressed by quality measures, seems to be the most obvious criterion for algorithm selection, the computational complexity is very often equally important. Therefore, in this Section, a straightforward analysis of computational complexity of the Fast Adaptive Switching Trimmed filter with AMF output—FASTAMF is presented. This filter was chosen as it belongs to the fastest available filtering designs and because of its very satisfying denoising efficiency. It will be compared with the state-of-the-art fast techniques: FPGF [34], Fast Averaging Peer Group Filter (FAPGF) [36], Fast Fuzzy Noise Reduction Filter (FFNRF) [56], Fast Modified VMF [62] and Vector Median Filter (VMF) [6] which can serve as a reference filter. The analysis of the computational burden will be performed for impulse detection (decision-making step) and output computation step separately.

We assume that color image is encoded with L channels, and the operating window W used by the filter consists of n pixels. The elementary mathematical operations used by an

algorithm will be labeled as follows: Addition—ADD, Multiplication—MULT, Division—DIV, Exponentiation—EXP, Extractions of root—SQRT, Comparison—COMP. A detailed analysis of the computational load with commentary is performed for FASTAMF algorithm only. The complexity of the competitive algorithms is summarized in Table 7.

The impulse detection step of the FASTAMF requires:

- Computation of $(n - 1)$ Euclidean distances. Each distance requires: $L \times \text{MULT} + 2L \times \text{ADD} + 1 \times \text{SQRT}$.
- Calculation of the m smallest distances:

$$m \sum_{i=1}^m (n - i) \times \text{COMP}, \quad (10)$$

- Sum of the m smallest distances: $m \times \text{ADD}$,
- One subtraction ($1 \times \text{ADDS}$), division ($1 \times \text{DIV}$ and comparison ($1 \times \text{COMP}$).

As during the noisy pixel replacement, the algorithm requires the *map of noise* array M , consisting of values: 1 for uncorrupted pixels, when condition (6) is satisfied and 0 for pixels found to be corrupted, the output computation step of the AMF requires $n \times L \times \text{MULT}$, $n \times L \times \text{ADDS}$ to acquire a sum of uncorrupted pixels channel values and $n \times \text{ADD}$, $1 \times \text{DIV}$ to obtain the final color pixel estimate.

Table 5 Comparison of the efficiency of the analyzed algorithms with the state-of-the-art techniques for the GOLDHILL color test image

Measure	p	New algorithms			State-of-the-art algorithms					
		STAMF	ASTAMF	FASTAMF	ACWVMF	FAPGF	FFNRF	FOVMF	FPGF	SVMFr
PSNR	0.1	37.43	37.76	37.71	35.67	36.80	35.47	30.53	34.89	34.00
	0.2	34.13	34.29	34.35	30.69	33.72	32.25	29.65	31.82	29.08
	0.3	32.00	31.28	32.11	26.15	31.66	28.53	27.88	28.34	23.93
	0.4	29.63	27.32	29.36	22.20	29.46	24.59	25.09	24.58	19.96
	0.5	26.60	23.04	25.51	18.90	26.64	20.72	21.77	20.93	16.78
NCD (E-04)	0.1	55.92	54.19	55.04	66.96	59.51	67.29	393.33	71.96	96.19
	0.2	115.01	112.58	113.34	159.99	122.13	142.38	420.19	146.05	191.84
	0.3	178.36	187.82	177.84	312.77	195.13	254.94	473.14	250.51	415.14
	0.4	259.01	326.21	269.56	580.28	291.17	450.57	589.60	428.73	825.61
	0.5	386.77	630.09	460.10	1027.57	456.95	819.96	840.39	758.13	1497.38
MAE	0.1	0.70	0.68	0.68	0.82	0.75	0.84	4.81	0.97	1.26
	0.2	1.43	1.40	1.40	1.87	1.51	1.64	5.25	1.91	2.21
	0.3	2.21	2.26	2.17	3.58	2.37	2.84	6.00	3.27	4.48
	0.4	3.21	3.71	3.22	6.54	3.48	4.84	7.46	5.51	8.62
	0.5	4.76	6.71	5.19	11.43	5.24	8.64	10.49	9.51	15.43
FSIMc	0.1	0.9966	0.9968	0.9968	0.9788	0.9961	0.9787	0.9671	0.9777	0.9762
	0.2	0.9916	0.9920	0.9921	0.9677	0.9909	0.9734	0.9611	0.9702	0.9621
	0.3	0.9853	0.9825	0.9860	0.9422	0.9841	0.9603	0.9476	0.9532	0.9243
	0.4	0.9738	0.9576	0.9731	0.8903	0.9711	0.9278	0.9139	0.9126	0.8578
	0.5	0.9506	0.9015	0.9381	0.8187	0.9449	0.8659	0.8556	0.8465	0.7750

Bold values represent the best results obtained (in a corresponding row)

As can be derived from Table 7, the proposed FASTAMF algorithm belongs to the group of the fast switching filters and its computational complexity is comparable only to FPGF [34] and newly proposed FAPGF technique [36].

For the assessment of the practical usability of the proposed filtering framework in real-time applications, we took measurements of the processing time using a large MOSAIC test image of size 3200×2400 pixels depicted in Fig. 14. To make the results independent on the structural content of the processed data and also to present the acceleration achieved using the CUDA parallel programming platform, this benchmark image is composed of 25 pictures taken from the database shown in Fig. 3.

The first group of tests was performed on a machine equipped with Intel i7-3632QM processor unit (2.2 GHz) and 4 GB memory. The 64-bit Debian 8.3 was installed as the operating system. For the purpose of the speed tests, all examined filtering techniques were implemented in ANSI C (gcc 4.9.2) programming language. To assure the fairness of the results and to avoid inefficient algorithm implementations, all of them with the exception of FASTAMF and FAPGF, which we prepared ourselves, were taken from the “Fourier 0.8” library provided by M.E. Celebi [63]. The speed measurements were also taken using a routine from this well-known library.

All comparative tests of the analyzed filters were performed using single-thread processing. The filtering window was consistently 3×3 , and the parameters of the evaluated techniques were adjusted according to the recommendations of their authors. Each filter was run 200 times on the MOSAIC test image, contaminated with impulsive noise of intensity $p = 0, 0.1, \dots, 0.5$, to assure the statistical significance of the comparisons. The medians of the independent execution times of all tested filters are presented in Table 8 and Fig. 15. The FOVMF algorithm was omitted in the presentation of results due to its very poor performance.

As can be observed, the proposed FASTAMF method is about 3 times faster than the standard VMF. The execution time of the FASTAMF measured on the MOSAIC image with medium noise level was about 800 ms. The time needed to process a 640×480 image was on average 34 ms, which confirms that the computational complexity grows linearly with the number of image pixels. The processing time of FASTAMF is only slightly dependent on the noise intensity, and this behavior is also exhibited by other filtering approaches, with the exception of FPGF, which is the fastest filter for very low contamination ratios ($p < 0.1$), but is slowing down substantially with increasing noise corruption.

Table 6 Comparison of the efficiency of the analyzed algorithms with the state-of-the-art techniques for the FLOWER color test image

Measure	p	New algorithms			State-of-the-art algorithms					
		STAMF	ASTAMF	FASTAMF	ACWVMF	FAPGF	FFNRF	FOVMF	FPGF	SVMFr
PSNR	0.1	38.57	38.92	38.85	36.81	37.71	37.10	33.30	36.04	36.03
	0.2	35.71	36.03	36.01	31.90	35.06	33.26	31.58	32.80	30.20
	0.3	33.06	32.42	33.28	27.21	32.59	29.22	29.23	29.32	25.01
	0.4	30.54	28.26	30.59	23.35	30.30	25.29	26.32	25.73	21.00
	0.5	27.36	23.87	26.50	20.13	27.19	21.57	23.15	22.29	17.79
NCD (E-04)	0.1	30.53	28.85	29.42	38.09	34.08	35.47	172.10	42.08	45.83
	0.2	62.45	59.22	59.43	97.16	70.27	82.74	206.74	87.92	117.66
	0.3	102.41	108.78	98.45	211.51	120.56	163.68	261.49	160.42	286.79
	0.4	155.46	212.72	156.04	410.70	192.40	314.86	360.95	286.26	610.58
	0.5	248.66	460.79	295.13	748.21	326.95	604.49	555.01	517.64	1149.86
MAE	0.1	0.46	0.45	0.45	0.53	0.51	0.49	2.50	0.66	0.67
	0.2	0.92	0.88	0.89	1.28	1.02	1.07	3.09	1.34	1.47
	0.3	1.50	1.51	1.44	2.64	1.68	2.02	3.91	2.40	3.30
	0.4	2.24	2.68	2.19	4.93	2.56	3.70	5.17	4.07	6.72
	0.5	3.47	5.26	3.77	8.72	4.10	6.81	7.44	6.94	12.42
FSIMc	0.1	0.9975	0.9977	0.9976	0.9799	0.9967	0.9803	0.9723	0.9785	0.9791
	0.2	0.9941	0.9947	0.9945	0.9721	0.9932	0.9758	0.9657	0.9723	0.9686
	0.3	0.9891	0.9877	0.9899	0.9510	0.9867	0.9639	0.9535	0.9580	0.9387
	0.4	0.9788	0.9656	0.9796	0.9121	0.9747	0.9376	0.9304	0.9306	0.8836
	0.5	0.9605	0.9169	0.9522	0.8530	0.9512	0.8870	0.8886	0.8824	0.8071

Bold values represent the best results obtained (in a corresponding row)

In Fig. 16, we present the relation between the efficiency of the evaluated algorithms expressed in terms of PSNR and the processing time obtained for the MOSAIC test image contaminated by impulsive noise of $p = 0.1$ and $p = 0.3$. Analyzing the results presented in the plots and also in Table 8 and Fig. 15, it can be stated that the proposed FASTAMF is one of the most efficient algorithms among those taken for comparisons in the whole range of contamination ratios and its overall efficiency is comparable with the FAPGF and also with FPGF when the noise intensity is quite low.

Although for low contamination level the proposed FASTAMF is slower than FPGF, its efficiency expressed in terms of PSNR and other quality measures is significantly better. The FASTAMF technique is generally slightly slower than the FAPGF, but for low and medium noise corruption it is generally superior in terms of the denoising efficiency.

The new technique can be further accelerated by omitting the computation of already determined distances between pixels. Such a scheme was successfully applied in the construction of fast VMF implementations [64, 65]. Further substantial decrease in the computational time can be achieved using the hardware/software implementations, which are being developed very rapidly especially for image processing applications [66–68].

In our algorithm, the image is processed in 3 steps which must be performed sequentially: computing of trimmed sum of distances (3), noise detection (6) and noisy pixel replacement. However, every pixel of the image can be processed independently during particular step. Therefore, each of those steps can be implemented as parallel processing which substantially decreases their execution times.

We implemented the VMF and FASTAMF using the CUDA technology on GeForce GTX 970 GPU equipped with 1664 CUDA cores (1250 MHz) and 4 GB 256-bit GDDR5 memory. The FASTAMF algorithm was written in C++ and compiled under NVIDIA CUDA compiler.

The grid configurations were chosen dynamically depending on the image size. We have chosen a block size of 128 threads in a configuration $1 \times 128 \times 1$ threads. The crucial part is the right selection of the grid and block configuration, depending of the size of the image and the GPU parameters, so that the GPU computation ability is maximized.

The second essential part is the optimization of memory reads and writes. In the first step, we copy the image from host into the GPU memory. In that way, we minimize the use of the slow throughput between host and device. Another optimization is the correct way of reading global

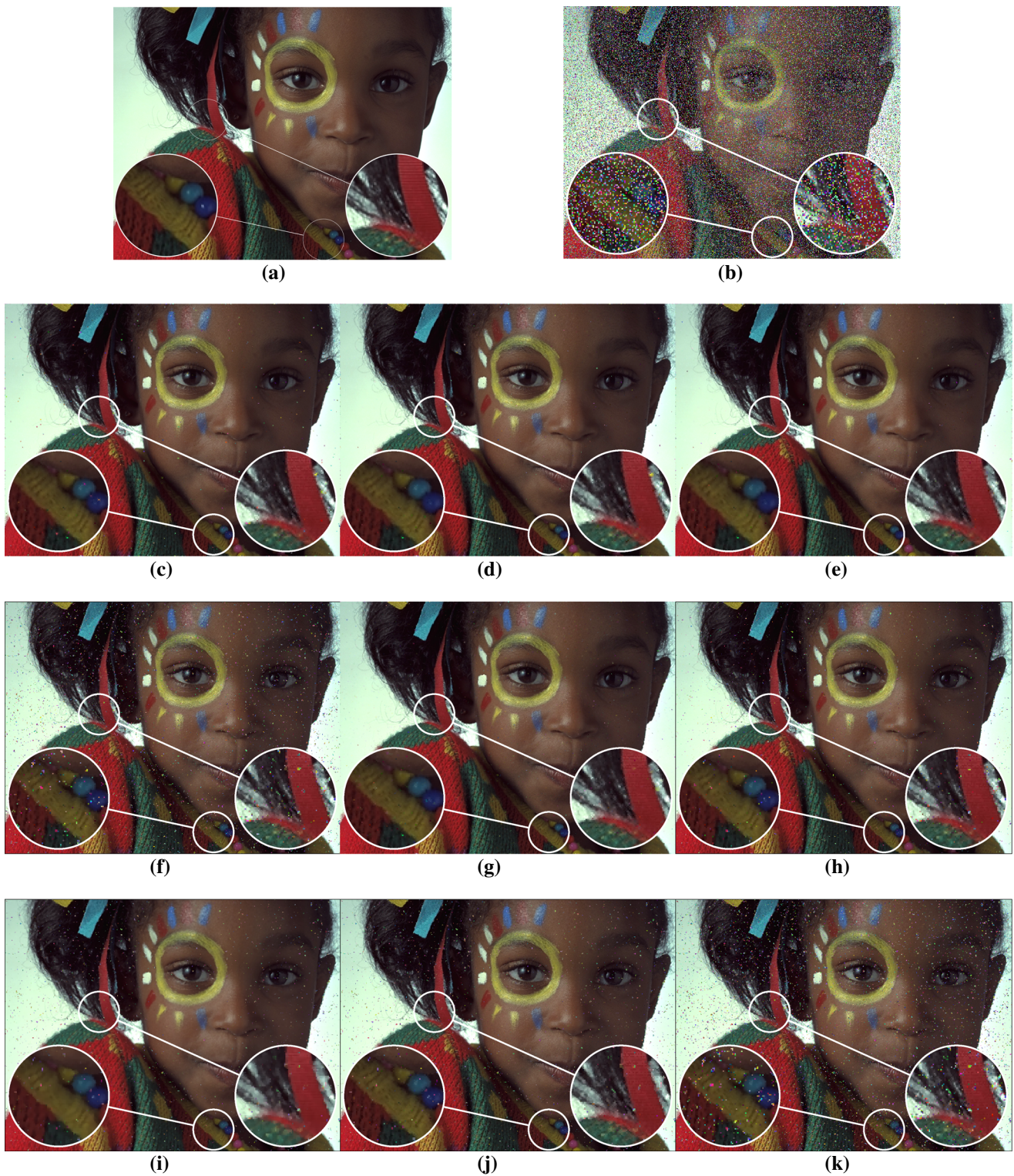


Fig. 11 Comparison of the performance of the filtering algorithms for image GIRL contaminated with impulsive noise of intensity $p = 0.3$. **a** Original image. **b** Noisy image. **c** ASTAMF. **d** FASTAMF. **e** STAMF. **f** ACWVMF. **g** FAPGF. **h** FFNRF. **i** FOVMF. **j** FPGF. **k** SVMFr

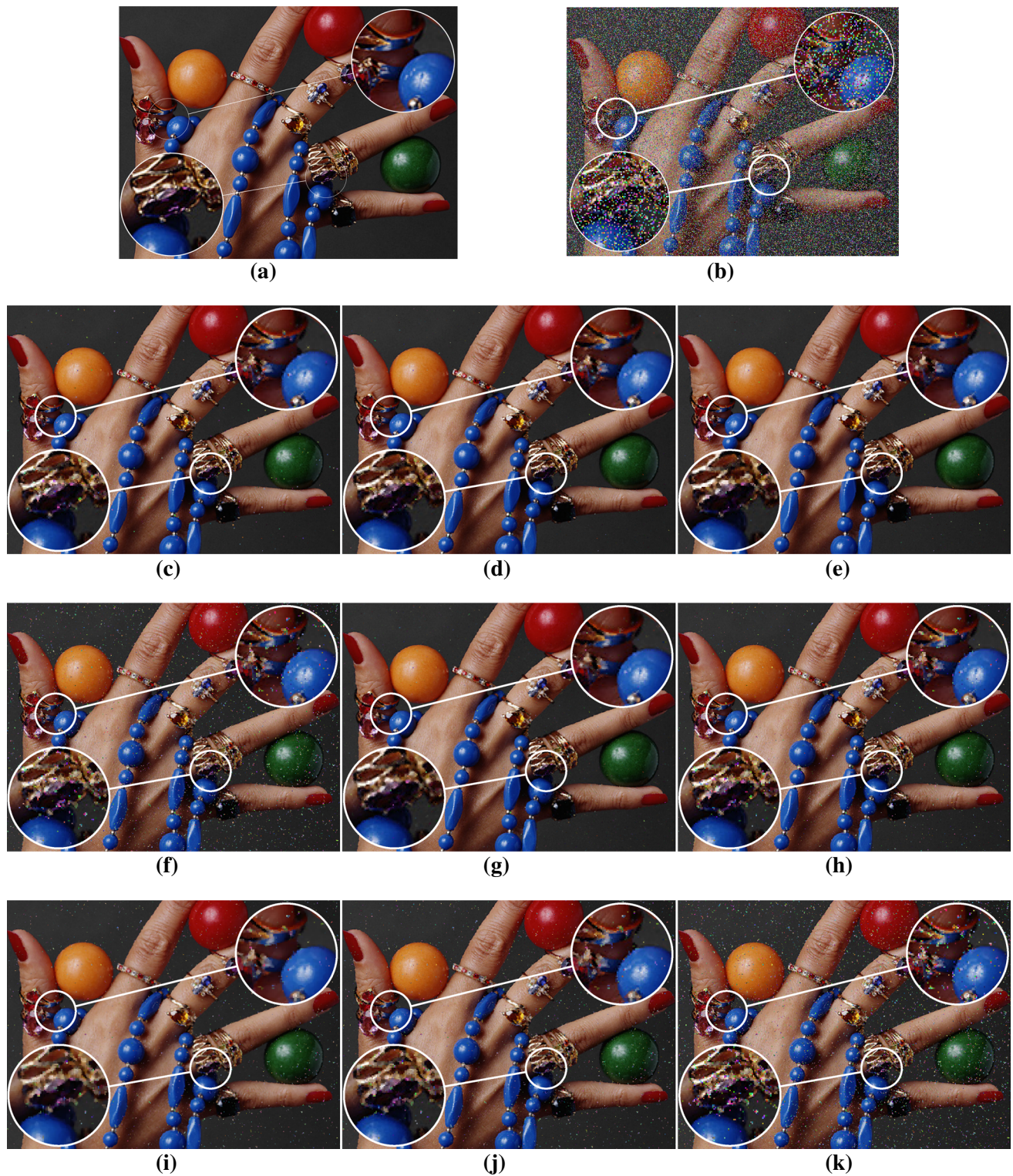


Fig. 12 Comparison of the performance of the filtering algorithms for image HAND contaminated with impulsive noise of intensity $p = 0.3$. **a** Original image. **b** Noisy image. **c** ASTAMF. **d** FASTAMF. **e** STAMF. **f** ACWVMF. **g** FAPGF. **h** FFNRF. **i** FOVMF. **j** FPGF. **k** SVMFr

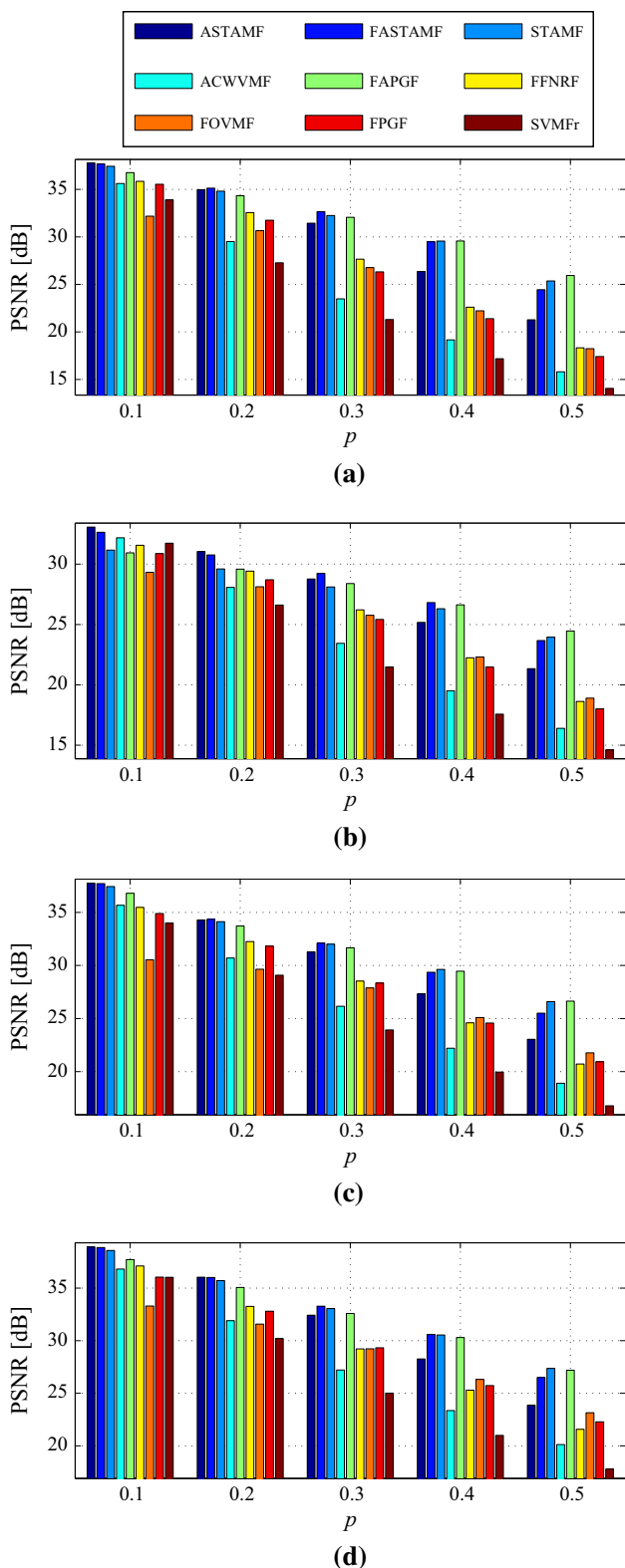


Fig. 13 Comparison of the proposed designs with state-of-the-art denoising algorithms using four test images contaminated with noise of intensity $p = 0.1, \dots, 0.5$. **a** GIRL. **b** HAND. **c** GOLDHILL. **d** FLOWER

memory, which is the slowest one on the device, but is accessible by each thread. We use optimal access patterns based on the GPU computational capability and utilize data types that meet the size and alignment which is optimal for the given device.

Three kernels, which were run successively on each image pixel, were implemented, and the number of reads and writes from the global memory was reduced to minimum.

The speed tests of CUDA FASTAMF implementation were performed on the MOSAIC image, and the results are presented in Fig. 17. It can be observed that using parallel computing, impressive speed gains can be achieved, which allows to use the FASTAMF in real-time image processing. As only a few milliseconds are needed to process the relatively large MOSAIC image (3200×2400), the algorithm can be applied for video denoising with frame rates exceeding 100 fps or much more for smaller resolutions.

6 Conclusions

The evaluation of the performance of the described filter family provided in the previous Sections confirmed its high efficiency. The proposed filters are competitive against known fast filtering techniques intended for impulsive noise removal. Especially useful is the Fast Adaptive Switching Trimmed filter with AMF output—FASTAMF, which restores efficiently the corrupted pixels even for strong noise contamination. Its performance is comparable with the recently proposed Fast Averaging Peer Group Filter (FAPGF) [36]. The beneficial feature of the FASTAMF is its low computational complexity, which makes the filter interesting for the real-time color image denoising.

The proposed concept of trimmed sum of ordered distances is a very efficient way of determining whether a pixel is corrupted or not. Also the AMF output computed using only pixels recognized as uncorrupted, proved to be a very efficient and computationally inexpensive solution.

Additionally, the adaptation mechanism implemented in the AST and FAST decision-making schemes substantially improves the performance of the filters when the image is contaminated by noise of low and medium intensity ($p < 0.3$). For higher noise intensity levels, this mechanism fails to detect the outliers, due to a small number of the uncorrupted samples in the filtering window.

The performed experiments confirmed the low computational complexity of the proposed filtering technique and its attractiveness for real-time image processing

Table 7 Comparison of computational complexity

Algorithm	ADDS	MULTS	DIVS	EXPS	SQRTS	COMPS	TOTAL
Impulse detection for $L = 3, n = 3$							
FASTAMF ($m = 2$)	26	24	1	0	8	26	85
FAPGF	32	24	0	0	8	8	72
FPGF	32	24	0	0	8	8	72
VMF	0	0	0	0	0	0	0
FFNRF	0	0	0	0	0	0	0
FMVMF	186	63	0	0	21	8	278
Output computation for $L = 3, n = 3$							
FASTAMF ($m = 2$)	24	0	3	0	0	10	38
FAPGF	27	27	3	9	0	1	68
FPGF	186	63	0	0	21	8	278
VMF	186	63	0	0	21	8	278
FFNRF	48	24	24	24	0	55	175
FMVMF	0	0	0	0	0	1	1
Impulse detection for $L = 3, n = 5$							
FASTAMF ($m = 2$)	74	72	1	0	24	74	245
FAPGF	96	72	0	0	24	24	216
FPGF	96	72	0	0	24	24	216
VMF	0	0	0	0	0	0	0
FFNRF	0	0	0	0	0	0	0
FMVMF	855	330	0	0	110	24	1319
Output computation for $L = 3, n = 5$							
FASTAMF ($m = 2$)	72	0	3	0	0	26	101
FAPGF	75	75	3	25	0	1	179
FPGF	855	330	0	0	110	24	1319
VMF	855	330	0	0	110	24	1319
FFNRF	144	72	72	72	0	167	527
FMVMF	0	0	0	0	0	1	1

Table 8 Processing times of the filters taken for comparisons when restoring the MOSAIC test image depicted in Fig. 14 contaminated with impulsive noise of increasing intensity p

p	VMF	FASTAMF	ACWVMF	FAPGF	FFNRF	FPGF	SVMFr	FOVMF
0.0	2.460	0.748	2.986	0.649	1.328	0.344	2.481	7.167
0.1	2.460	0.797	3.034	0.698	1.387	0.632	2.504	7.278
0.2	2.460	0.847	3.068	0.752	1.431	0.923	2.522	7.260
0.3	2.460	0.905	3.094	0.816	1.476	1.229	2.541	7.241
0.4	2.460	0.971	3.117	0.887	1.525	1.558	2.557	7.209
0.5	2.460	1.024	3.133	0.957	1.565	1.905	2.569	7.220

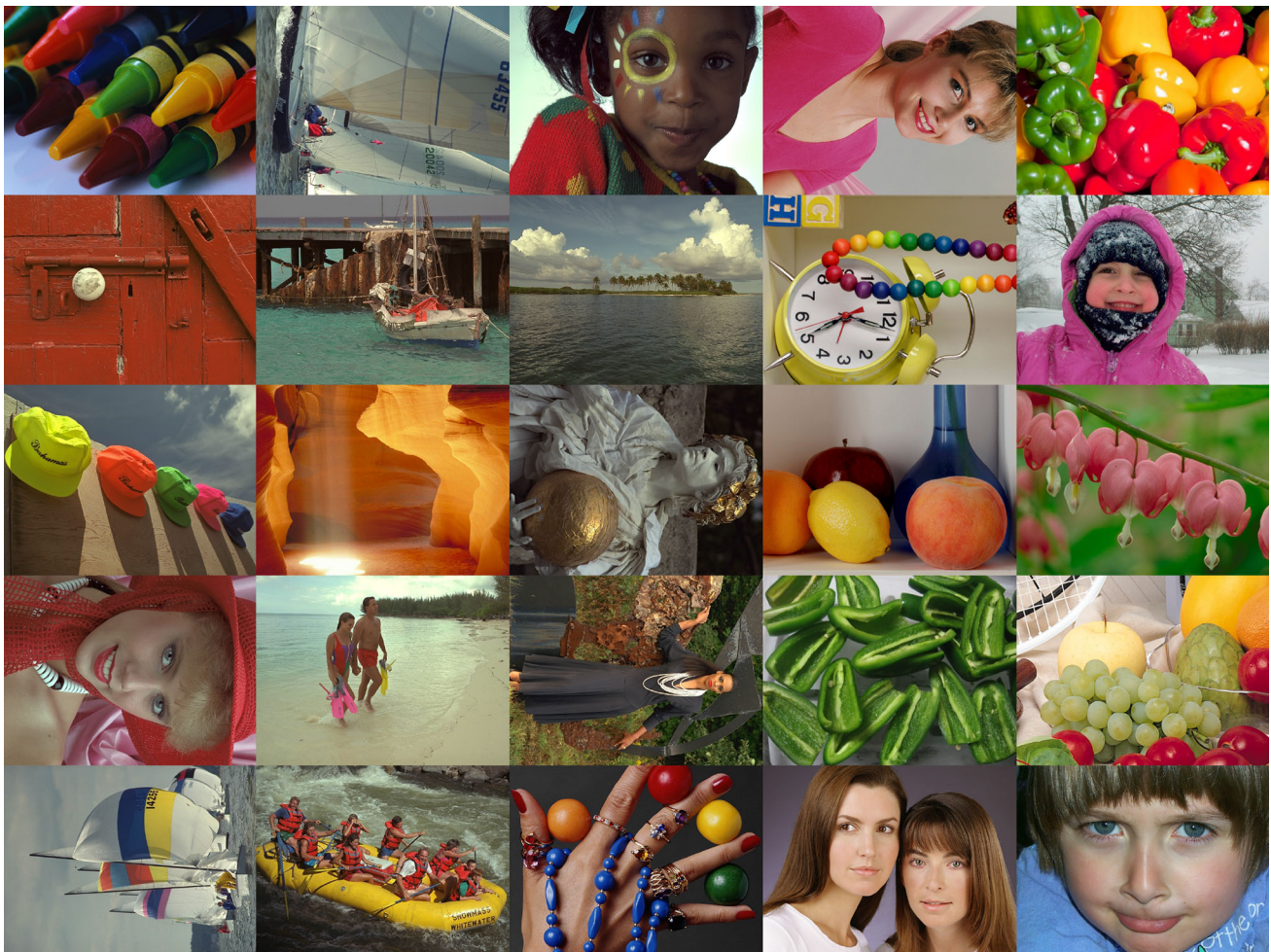


Fig. 14 Test image MOSAIC of size 3200×2400 pixels composed of 25 pictures from the dataset depicted in Fig. 3

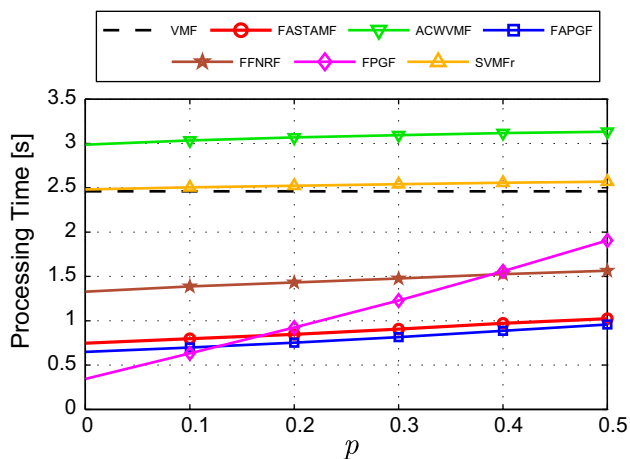


Fig. 15 Processing times of the evaluated filters using the MOSAIC test image depicted in Fig. 14 contaminated with impulsive noise of intensity p

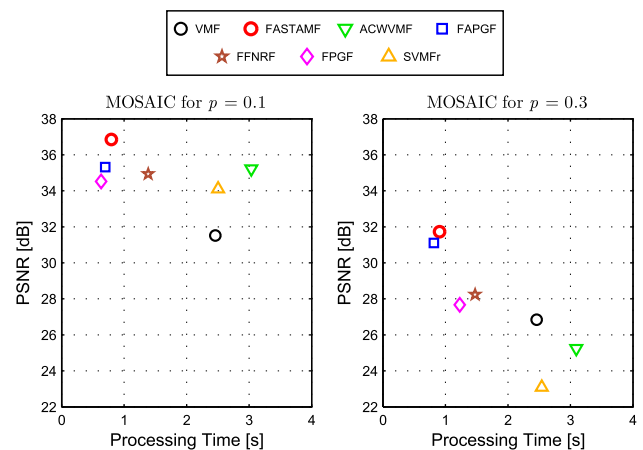


Fig. 16 Relation between PSNR measure and the processing time of the evaluated filters obtained using the MOSAIC test image contaminated with impulsive noise of intensity $p = 0.1$ and $p = 0.3$

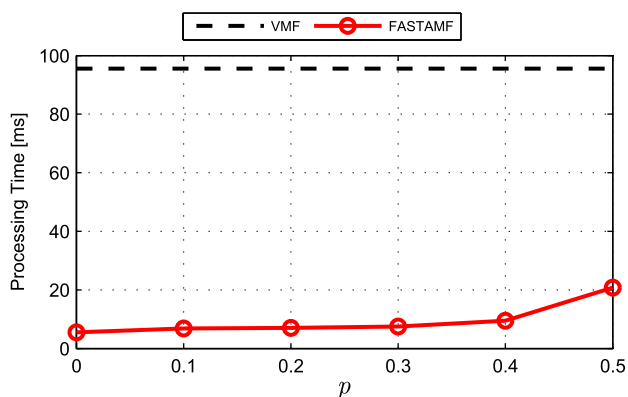


Fig. 17 Comparison of the FASTAMF and VMF execution time performed on the noisy MOSAIC test image using the CUDA implementations

applications. Additional speed gain was obtained using a parallel implementation on the CUDA platform, which allows to apply the proposed algorithm for video denoising.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Plataniotis, K., Venetsanopoulos, A.: *Color Image Processing and Applications*. Springer, Berlin (2000)
- Boncellet, C.G.: Image noise models. In: Bovik, A.C. (ed.) *Handbook of Image and Video Processing, Communications, Networking and Multimedia*, pp. 397–410. Academic Press, London (2005)
- Lukac, R., Smolka, B., Martin, K., Plataniotis, K., Venetsanopoulos, A.: Vector filtering for color imaging. *IEEE Signal Process. Mag.* **22**(1), 74–86 (2005)
- Zheng, J., Valavanis, K.P., Gauch, J.M.: Noise removal from color images. *J. Intell. Robot. Syst.* **7**(1), 257–285 (1993)
- Morillas, S., Gregori, V., Sapena, A., Camarena, J., Roig, B.: Impulsive noise filters for colour images. In: Celebi, M.E., Lecca, M., Smolka, B. (eds.) *Color Image and Video Enhancement*. Springer, Berlin (2015)
- Astola, J., Haavisto, P., Neuvo, Y.: Vector median filters. *Proc. IEEE* **78**(4), 678–689 (1990)
- Nikolaidis, N., Pitas, I.: Multivariate ordering in color image processing. *Signal Process.* **38**(3), 299–316 (1994)
- Tang, K., Astola, J., Neuvo, Y.: Nonlinear multivariate image filtering techniques. *IEEE Trans. Image Process.* **4**(6), 788–798 (1995)
- Pitas, I., Tsakalides, P.: Multivariate ordering in color image processing. *IEEE Trans. Circuits Syst. Video Technol.* **1**(3), 247–256 (1991)
- Smolka, B., Plataniotis, K., Venetsanopoulos, A.: Nonlinear techniques for color image processing. In: Barner, K.E., Arce, G.R. (eds.) *Nonlinear Signal and Image Processing: Theory, Methods, and Applications*. CRC Press, Boca Raton (2004)
- Smolka, B., Venetsanopoulos, A.: Noise reduction and edge detection in color images. In: Lukac, R., Plataniotis, K.N. (eds.) *Color Image Processing: Methods and Applications*, pp. 75–100. CRC Press, Boca Raton (2006)
- Viero, T., Oistamo, K., Neuvo, Y.: Three-dimensional median-related filters for color image sequence filtering. *IEEE Trans. Circuits Syst. Video Technol.* **4**(2), 129–142 (1994)
- Ponomaryov, V., Gallegos-Funes, F., Rosales-Silva, A.: Real-time color image processing using order statistics filters. *J. Math. Imaging Vis.* **23**(3), 315–319 (2005)
- Smolka, B., Malik, K., Malik, D.: Adaptive rank weighted switching filter for impulsive noise removal in color images. *J. Real-Time Image Process.* **10**(2), 289–311 (2015). doi:[10.1007/s11554-012-0307-0](https://doi.org/10.1007/s11554-012-0307-0)
- Morillas, S., Gregori, V.: Robustifying vector median filter. *Sensors* **11**(8), 8115 (2011)
- Nair, M.S., Ameera Mol, P.M.: Direction based adaptive weighted switching median filter for removing high density impulse noise. *Comput. Electr. Eng.* **39**(2), 663–689 (2013)
- Lukac, R., Smolka, B., Plataniotis, K.N., Venetsanopoulos, A.N.: Selection weighted vector directional filters. *Comput. Vis. Image Underst.* **94**(1–3), 140–167 (2004)
- Lukac, R.: Adaptive vector median filtering. *Pattern Recognit. Lett.* **24**(12), 1889–1899 (2003)
- Smolka, B.: Peer group switching filter for impulse noise reduction in color images. *Pattern Recognit. Lett.* **31**(6), 484 (2010)
- Geng, X., Hu, X., Xiao, J.: Quaternion switching filter for impulse noise reduction in color image. *Signal Process.* **92**(1), 150–162 (2012)
- Jin, L., Li, D.: An efficient color impulse detector and its application to color images. *IEEE Signal Process. Lett.* **14**(6), 397–400 (2007)
- Morillas, S., Gregori, V., Peris-Fajarnés, G.: Isolating impulsive noise pixels in color images by peer group techniques. *Comput. Vis. Image Underst.* **110**(1), 102–116 (2008)
- Smolka, B., Lukac, R., Chydzinski, A., Plataniotis, K.N., Wojciechowski, W.: Fast adaptive similarity based impulsive noise reduction filter. *Real-Time Imaging* **9**(4), 261–276 (2003)
- Karakos, D.G., Trahanias, P.E.: Generalized multichannel image-filtering structures. *IEEE Trans. Image Process.* **6**(7), 1038–1045 (1997)
- Celebi, M., Kingravi, H., Aslandogan, Y.: Nonlinear vector filtering for impulsive noise removal from color images. *J. Electron. Imaging* **16**(3), 033008-1–033008-21 (2007)
- Celebi, M.E.: Real-time implementation of order-statistics based directional filters. *IET Image Process.* **3**(1), 1–9 (2009)
- Smolka, B., Malik, K.: Reduced ordering technique of impulsive noise removal in color images. In: Tominaga, S., Schettini, R., Trémeau, A. (eds.) *Computational Color Imaging. Lecture Notes in Computer Science*, vol. 7786, pp. 296–310. Springer, Berlin (2013)
- Lukac, R., Smolka, B., Plataniotis, K.N.: Sharpening vector median filters. *Signal Process.* **87**, 2085–2099 (2007)
- Garnett, R., Huegerich, T., Chui, C., He, W.: A universal noise removal algorithm with an impulse detector. *IEEE Trans. Image Process.* **14**(11), 1747–1754 (2005)
- Lukac, R., Smolka, B., Plataniotis, K.N., Venetsanopoulos, A.N.: Vector sigma filters for noise detection and removal in color images. *J. Vis. Commun. Image Represent.* **17**(1), 1–26 (2006)
- Lukac, R., Plataniotis, K.N., Venetsanopoulos, A.N., Smolka, B.: A statistically-switched adaptive vector median filter. *J. Intell. Robot. Syst.* **42**(4), 361–391 (2005)
- Deng, Y., Kenney, C., Manjunath, B.S.: Peer group image enhancement. *IEEE Trans. Image Process.* **10**(2), 326–334 (2001)

33. Smolka, B., Plataniotis, K.N., Chydzinski, A., Szczepanski, M., Venetsanopoulos, A.N., Wojciechowski, K.: Self-adaptive algorithm of impulsive noise reduction in color images. *Pattern Recognit.* **35**(8), 1771–1784 (2002)
34. Smolka, B., Chydzinski, A.: Fast detection and impulsive noise removal in color images. *Real-Time Imaging* **11**(5–6), 389–402 (2005)
35. Morillas, S., Gregori, V., Hervás, A.: Fuzzy peer groups for reducing mixed Gaussian-impulse noise from color images. *IEEE Trans. Image Process.* **18**(7), 1452–1466 (2009)
36. Malinski, L., Smolka, B.: Fast averaging peer group filter for the impulsive noise removal in color images. *J. Real-Time Image Process.* (2015). doi:[10.1007/s11554-015-0500-z](https://doi.org/10.1007/s11554-015-0500-z)
37. Jin, L., Li, D.: An efficient color impulse detector and its application to color images. *IEEE Signal Process. Lett.* **14**(6), 397–400 (2007)
38. Wang, G., Liu, Y., Zhao, T.: A quaternion-based switching filter for colour image denoising. *Signal Process.* **102**, 216–225 (2014)
39. Schulte, S., De Witte, V., Nachtegaele, M., Van der Weken, D., Kerre, E.E.: Fuzzy random impulse noise reduction method. *Fuzzy Sets Syst.* **158**, 270–283 (2007)
40. Varghese, J., Ghouse, M., Subash, S., Siddappa, M., Khan, M.S., Hussain, O.B.: Efficient adaptive fuzzy-based switching weighted average filter for the restoration of impulse corrupted digital images. *IET Image Process.* **8**(4), 199–206 (2014)
41. Kang, C., Wang, W.: Fuzzy reasoning-based directional median filter design. *Signal Process.* **89**(3), 344–351 (2009)
42. Melange, T., Nachtegaele, M., Kerre, E.: Fuzzy random impulse noise removal from color image sequences. *IEEE Trans. Image Process.* **20**(4), 959–970 (2011)
43. Ponomaryov, V., Montenegro, H., Rosales, A., Duchon, G.: Fuzzy 3D filter for color video sequences contaminated by impulsive noise. *J. Real Time Image Process.* **10**, 313–328 (2012)
44. Morillas, S., Gregori, V., Peris-Fajarnés, G., Latorre, P.: A fast impulsive noise color image filter using fuzzy metrics. *Real-Time Imaging* **11**(5–6), 417–428 (2005)
45. Hore, E.S., Qiu, Bin, Wu, H.R.: Improved vector filtering for color images using fuzzy noise detection. *Opt. Eng.* **42**(6), 1656–1664 (2003)
46. Smolka, B.: Robustified vector median filter. In: 9th International Conference on Computer Science Education (ICCSE), 2014, pp. 362–367, Aug 2014
47. Smolka, B., Andrzejczak, A., Nabialkowski, P., Nelip, A.: Thresholded median filter for the impulsive noise removal in digital images. In: The 5th International Conference on Information, Intelligence, Systems and Applications (IISA'2014), pp. 355–360, July 2014
48. Smolka, B.: Fast impulsive noise removal in color images. In: IEEE International Conference on Image Processing (ICIP'2013) Melbourne, Australia, pp. 1212–1216 (2013)
49. Phu, M.Q., Tischer, P.E., Wu, H.R.: Statistical analysis of impulse noise model for color image restoration. In: 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS'2007) 2007
50. Hamza, A.B., Krim, H.: Image denoising: a nonlinear robust statistical approach. *IEEE Trans. Signal Process.* **49**(12), 3045–3054 (2001)
51. Pilevar, A.H., Saien, S., Khandel, M., Mansoori, B.: A new filter to remove salt and pepper noise in color images. *SIViP* **9**(4), 779–786 (2015). ISSN: 1863-1703
52. Wang, G., Li, D., Pan, W., Zang, Z.: Modified switching median filter for impulse noise removal. *Signal Process.* **90**(12), 3213–3218 (2010)
53. Esakkirajan, S., Veerakumar, T., Subramanyam, A.N., Premchand, C.H.: Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. *IEEE Signal Process. Lett.* **18**(5), 287–290 (2011)
54. Venkatesan, P., Nagarajan, G.: Removal of Gaussian and impulse noise in the colour image progression with fuzzy filters. *Int. J. Electron. Signals Syst.* **3**(1), 1–6 (2013)
55. Lukac, R.: Adaptive color image filtering based on center-weighted vector directional filters. *Multidimens. Syst. Signal Process.* **15**(2), 169–196 (2004)
56. Morillas, S., Gregori, V., Peris-Fajarnés, G., Latorre, P.: A new vector median filter based on fuzzy metrics. In: Kamel, M., Campilho, A. (eds.) *Image Analysis and Recognition. Lecture Notes in Computer Science*, vol. 3656, pp. 81–90. Springer, Berlin (2005)
57. Plataniotis, K.N., Androutsos, D., Venetsanopoulos, A.N.: Adaptive fuzzy systems for multichannel signal processing. *Proc. IEEE* **87**(9), 1601–1622 (1999)
58. Zhang, L., Zhang, L., Mou, X., Zhang, D.: FSIM: a feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **20**(8), 2378–2385 (2011)
59. Mou X., Zhang, L., Zhang, L., Zhang, D.: FSIM: a feature similarity index for image quality assessment, (2013). <http://sse.tongji.edu.cn/linzhang/IQA/FSIM/FSIM.htm>
60. Wang, Z., Bovik, A., Sheikh, A., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
61. Lee, D., Plataniotis, K.N.: Towards a full-reference quality assessment for color images using directional statistics. *IEEE Trans. Image Process.* **24**(11), 3950–3965 (2015)
62. Smolka, B., Szczepanski, M., Plataniotis, K.N., Venetsanopoulos, A.N.: Fast modified vector median filter. In: Skarbek, W. (ed.) *Computer Analysis of Images and Patterns. Lecture Notes in Computer Science*, vol. 2124, pp. 570–580. Springer, Berlin (2001)
63. Celebi, M.E.: Fourier 0.8, (2008). <http://sourceforge.net/projects/fourier-ipal>
64. Stolinski, S., Grabowski, S., Bieniecki, W.: On efficient implementation of median filters in theory and in practice. *Automatyka* **13**(3), 1021–1032 (2009)
65. Kim, J., Wills, D.S.: Fast vector median filter implementation using the color pack instruction set. In: Proceedings of 2002 IEEE 10th Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop, pp. 339–343, Oct 2002
66. Boudabous, A., Ben Atitallah, A., Kadionik, P., Khrijji, L., Masmoudi, N.: HW/SW FPGA implementation of vector median filter. In: Research in Microelectronics and Electronics Conference, 2007. (PRIME'2007). Ph.D., pp. 101–104, July 2007
67. Sánchez, M.G., Vidal, V., Bataller, J., Arnal, J.: A parallel method for impulsive image noise removal on hybrid CPU/GPU systems. In: 2013 International Conference on Computational Science. *Procedia Computer Science*, vol. 18, pp. 2504–2507, 2013
68. Sánchez, M.G., Vidal, V., Arnal, J., Vidal, A.: Image noise removal on heterogeneous cpu-gpu configurations. In: 2014 International Conference on Computational Science. *Procedia Computer Science*, vol. 29, pp. 2219–2229, 2014

Lukasz B. Malinski is an Assistant Professor at the Division of Industrial Informatics. He graduated from Silesian University of Technology and received M.Sc. degree in Automatic Control And Robotics in June 2009. After that he started the Ph.D. studies and performed research in field of identification of bilinear time-series models. He received his Ph.D. in April 2014. His current main scientific interest is focused on stochastic optimisation for nonlinear problems, but also he explores fields of system identification and image processing. He is the author of over 17 articles.

Bogdan Smolka received the Diploma in Physics degree from the Silesian University, Katowice, Poland, in 1986 and the Ph.D. degree in computer science from the Department of Automatic Control, Silesian University of Technology, Gliwice, Poland, in 1998. From 1986 to 1989, he was a Teaching Assistant at the Department of Biophysics, Silesian Medical University, Katowice, Poland. From 1992 to 1994, he was a Teaching Assistant at the Technical University of Esslingen, Germany. Since 1994, he has been with the Silesian University of Technology. In 1998, he was appointed as

an Associate Professor in the Department of Automatic Control. He has also been an Associate Researcher with the Multimedia Laboratory, University of Toronto, Canada since 1999. In 2007, Dr. Smolka was promoted to Professor at the Silesian University of Technology. He has published over 200 papers on digital signal and image processing in refereed journals and conference proceedings. His current research interests include low-level color image processing, human-computer interaction, and visual aspects of image quality.