



# Self-tuning fast adaptive algorithm for impulsive noise suppression in color images

Lukasz Malinski<sup>1</sup> · Bogdan Smolka<sup>2</sup>

Received: 11 April 2018 / Accepted: 18 January 2019 / Published online: 14 February 2019  
© The Author(s) 2019

## Abstract

In this paper, a self-tuning version of the newly introduced Fast Adaptive Switching Trimmed Arithmetic Mean Filter, which is a very efficient technique for impulsive noise suppression, is elaborated. Most of the methods presented in the rich literature have numerous parameters, whose proper settings are crucial for efficient noise suppression. Although researchers often provide recommended values for their algorithms' parameters, the actual choice remains in the hands of the user. Our goal is to free the operator from parameter selection dilemma and to propose an algorithm which includes required expert knowledge within itself. The only obligatory inputs of the proposed algorithm (from the user perspective) are the image itself and the size of the operating window.

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

## 1 Introduction

Rapid development of miniaturized high-resolution, low-cost image sensors, dedicated to operate in various lighting conditions, makes image enhancement and noise suppression to be very important operations of digital image processing.

There are various types of noise which affect acquisition and processing of digital color images. The disturbances may be introduced by [1–6]:

- electric signal instabilities,
- physical imperfections in sensors,
- corrupted memory locations,
- transmission errors,
- aging of the storage material,
- natural or artificial electromagnetic interferences.

Therefore, noise suppression is one of the most frequently performed low-level image processing tasks [1, 2, 5, 6]. There are plentiful different techniques tailored for suppression of distinct type of noise, but most of them are vulnerable to occurrence of a impulsive noise, which introduces significant deviations of color image channel values [7–9]. Therefore, the suppression of the impulsive noise is a critical step of image preprocessing.

Impulsive noise removal techniques are contextual processing schemes which estimate the channels of the processed pixel using information obtained from its neighborhood, represented by a sliding operational window. Many of them are based on a vector-ordering scheme [10–14], and use cumulative distances between samples in a window as dissimilarity estimates. Those accumulated distances are then sorted and constitute the basis for further processing in various filtering algorithms.

One of the most basic filtering techniques, utilizing this ordering scheme, is the vector median filter (VMF) [10, 15]. The output of VMF is the pixel from operational window for which the sum of distances to other samples from the window is minimized. Although this filter does not introduce any new colors to the processed image, there is no guarantee that the output pixel is itself noise-free, and thus, numerous solutions were developed to solve this problem and improve filtering performance [16–21].

✉ Lukasz Malinski  
lukasz.malinski@polsl.pl  
Bogdan Smolka  
bogdan.smolka@polsl.pl

<sup>1</sup> Division of Industrial Informatics, Silesian University of Technology, Krasinskiego 8, 40-019 Katowice, Poland

<sup>2</sup> Institute of Automatic Control, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

The main reason, that the efficiency of vector-ordering schemes is limited, lies in processing of every image pixel, regardless whether it is corrupted or not. Unnecessary processing of noise-free pixels results in inevitable degradation of the image quality. To address this issue, a significant improvement has been made by introduction of more sophisticated *switching filters* [22–31], which focus on the restoration of corrupted pixels only.

The switching techniques use various approaches to determine if the processed pixel is corrupted or not. Then, only those classified as noisy are further processed by the output estimation algorithm. This way, not only the quality of output of restored image is preserved, but also a significant reduction of the computational cost is often achieved.

There are numerous techniques of noisy pixel detection to be found in the literature [32–34]. Those schemes can be categorized by the following families:

- schemes based on *reduced vector ordering* [14, 35–38];
- techniques using *peer group* concept [39–42];
- filters utilizing *quaterions* [43–45];
- methods based on *fuzzy set theory* [46–53].

The Fast Adaptive Switching Trimmed Arithmetic Mean Filter (FASTAMF), concerned in this paper, has been proposed recently [54] and is a very efficient technique from both noise suppression efficiency and computational cost point of view. The main practical drawback of the algorithm (which is common among alternatives) is the necessity of manual parameter adjustment to image noise contamination severity, to achieve its optimal performance. Therefore, the main goal of the research presented here is the introduction of a self-tuning mechanism, so that manual experimental choice of the main filter parameter (threshold) will no longer be required.

## 1.1 Notation

For the purpose of better readability of subsequent sections describing the concerned algorithm design, the following notations are introduced:

- $X$ —input image (corrupted),
- $x_{u,v}$ —input image pixel located at spatial coordinates  $(u, v)$ ,
- $\hat{X}$ —output image (restored),
- $\hat{x}_{u,v}$ —output image pixel located at  $(u, v)$ ,
- $O$ —original image (reference image),
- $o_{u,v}$ —original image pixel located at  $(u, v)$ ,
- $M$ —original map of noise acquired using artificial image contamination.
- $m_{u,v}$ —real state of pixel corruption located at  $(u, v)$  (0—noisy; 1—noise-free),

- $\hat{M}$ —final estimated map of noise acquired during *noise detection* phase.
- $\hat{m}_{u,v}$ —classification of pixel contamination located at  $u, v$  (0—noisy; 1—noise-free).
- $W$ —local operating window centered at  $x_{u,v}$ , containing pixels from direct 8-neighborhood,
- $x_i$ — $i$ th pixel of the local operating window  $W$  (the pixel  $x_1$  is the central pixel in  $W$ ),
- $w$ —size of  $W$  (odd integer),
- $n$ —number of pixels in  $W$  ( $n = w \times w = w^2$ ),
- $d(x_i, x_j)$ —distance between two pixels from  $W$ ,
- $\delta_i$ —distance between central pixel  $x_1$  and  $x_i \in W$ ,
- $\delta_{(r)}$ — $r$ th smallest distance among all  $\delta_i$  computed for the same  $W$ ,
- $c_{u,v}$ —sum of  $\alpha$  smallest distances, representing raw impulsiveness of particular  $x_{u,v}$ ,
- $W_c$ —window containing values of raw impulsiveness computed for every pixel from local neighborhood of currently processed pixel  $x_{u,v}$ ,
- $c_{\min}$ —smallest accumulated distance in  $W_c$  representing simple estimate of image structure,
- $s_{u,v}$ —corrected impulsiveness measure of pixel at position  $u, v$ ,
- $k$ —iteration number in self-tuning (ST) procedure,
- $l$ —iteration number in multiple run test,
- $t$ —threshold value (filter parameter),
- $t_k$ —threshold value adjusted in the  $k$ th iteration of the self-tuning,
- $AMF(W)$ —output of the Arithmetic Mean Filter,
- $\rho$ —true noise density used in artificial image contamination.
- $\hat{\rho}$ —estimated noise density obtained through *noise detection* phase,
- $\hat{\rho}_k$ —estimated noise density obtained in the  $k$ th iteration of self-tuning procedure,
- $\rho_R, \rho_G, \rho_B$ —probability of contamination of channels in RGB color space,
- $\rho_A$ —probability of contamination of all pixel channels at once,
- $\hat{M}_k$ —estimated map of noise obtained during  $k$ th iteration of self-tuning,
- $\mu, \nu$ —height and width of the image (in pixels),
- $\theta$ —number of pixels in  $X$  ( $\theta = \mu \times \nu$ ),
- $n_k$ —number of pixels designated as noisy during  $k$ th iteration of self-tuning,
- $p$ —probability of error in statistical reasoning (result of statistical test which allows to hold or reject a null hypothesis).
- $k_F$ —final number of iterations of self-tuning procedure, required to satisfy the convergence condition.

In above, three-dimensional arrays (e.g., images, operating windows) are denoted using emboldened capital letters, two-dimensional arrays (e.g., map of noise) are indicated as normal capital letters, vectors (like single pixel) are presented as emboldened lowercase characters, and, finally, scalars are represented in normal lowercase manner.

### 1.2 Impulsive noise models

In this paper, four different noise models are considered [55, 56]. In all of those models, the main parameter is the noise density ( $\rho$ ) expressed by the percentage of corrupted pixels in the processed image:

- Channel Together Random Impulse (CTRI)—if a pixel is noisy, all of its RGB channels are corrupted.
- Channel Independent Random Impulse (CIRI)—if a pixel is contaminated, the alteration of every channel is independent.
- Channel Correlated Random Impulse (CIRI)—if a pixel is contaminated, then the corruption of channels is correlated with fixed correlation coefficient.
- Custom Probability Random Impulse (CPRI)—if a pixel is contaminated, there is a fixed set of probabilities that single RGB channels are corrupted ( $\rho_R, \rho_G, \rho_B$ ) or that all channels are corrupted together ( $\rho_A = 1 - (\rho_R + \rho_G + \rho_B)$ ). The model does not take into account the corruption of two channels at once.

In all above models, contaminated pixel channel is represented by a random value taken from full encoding range:  $\langle 0, 255 \rangle$  (for 8-bit RGB image coding).

### 1.3 Performance measures

The noise detection efficiency alone can be evaluated using binary classification. The result of noise detection, represented by estimated noise map  $\hat{M}$ , is compared to the original noise map  $M$ , established during artificial image corruption, which is treated as true information of noise occurrence. After comparison, each pixel can be assigned to one of the following classes:

- True positive (TP)—pixel was correctly recognized as being contaminated.
- False positive (FP)—pixel was falsely classified as noisy—also known as Type-I error.
- True negative (TN)—pixel was correctly recognized as not corrupted.
- False negative (FN)—pixel was incorrectly classified as not noisy—also known as Type-II error.

After an assignment of above states to every pixel in the image, the detection performance can be measured using accuracy:

$$ACC = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|}, \tag{1}$$

where  $|TP|$ ,  $|TN|$ ,  $|FP|$  and  $|FN|$  are cardinalities of pixels assigned to particular categories.

The overall noise suppression efficiency can be evaluated using many different performance measures. In this paper, we consider the following:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, \tag{2}$$

$$MSE = \frac{1}{3\theta} \sum_{u=1}^{\mu} \sum_{v=1}^{\nu} \sum_{q \in Z} (o_{u,v}^q - \hat{x}_{u,v}^q)^2, \tag{3}$$

where  $o_{u,v}^q$  and  $Z = \{R, G, B\}$  are the channels of the original image pixels, and  $\hat{x}_{u,v}^q$  are the pixels of the restored image:

$$MAE = \frac{1}{3\theta} \sum_{u=1}^{\mu} \sum_{v=1}^{\nu} \sum_{q \in Z} |o_{u,v}^q - \hat{x}_{u,v}^q|, \tag{4}$$

$$NCD = \frac{\sum_{u=1}^{\mu} \sum_{v=1}^{\nu} \sqrt{\Delta E}}{\sum_{u=1}^{\mu} \sum_{v=1}^{\nu} \sqrt{L_{u,v}^2 + a_{u,v}^2 + b_{u,v}^2}}, \tag{5}$$

$$\Delta E = (L_{u,v} - \hat{L}_{u,v})^2 + (a_{u,v} - \hat{a}_{u,v})^2 + (b_{u,v} - \hat{b}_{u,v})^2, \tag{6}$$

where  $L, a, b$  are the coordinates of the original and  $\hat{L}, \hat{a}, \hat{b}$  of restored image pixels, both in CIE Lab color space [1].

In addition, the Feature-SIMilarity index for color images (FSIMc), [57] was used to provide additional information about noise suppression performance. In contrast to Peak Signal-to-Noise Ratio (PSNR), Mean Absolute Error (MAE), and Normalized Color Difference (NCD), which operate on individual pixels, thus compare images in context-free manner, the FSIMc index, is based on the properties of the human visual system.

## 2 Original FASTAMF algorithm

The FASTAMF [54] algorithm is composed of two main processing phases (Fig. 1):

1. *noise detection*—the noise map ( $\hat{M}$ ) is estimated upon input image ( $X$ ), using reduced ordering scheme and

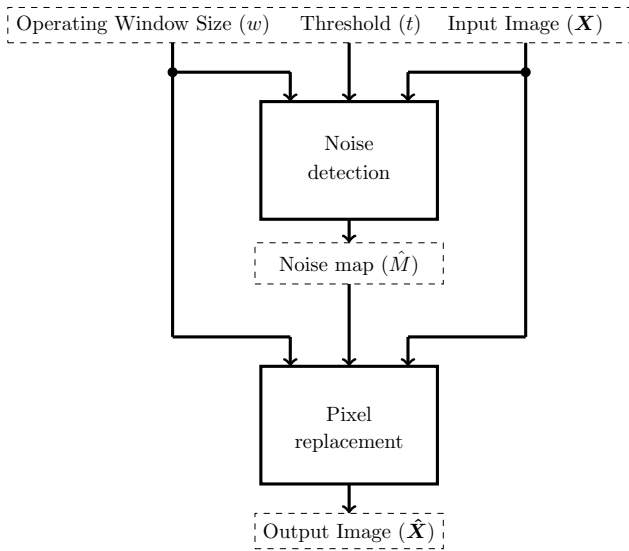


Fig. 1 Block diagram of original FASTAMF

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

Fig. 2 Notation of the pixels in the filtering window

two parameters provided by the user: operating window size ( $w$ ) and threshold ( $t$ ).

2. *pixel replacement*—the output image ( $\hat{X}$ ) is obtained using input image ( $X$ ), and noise map ( $\hat{M}$ ) provided by noise detection phase. Only pixels classified as noisy are processed by AMF with operating window size  $w$ .

The filter operates on every pixel of input image  $X$  located at coordinates  $(u, v)$ , denoted as  $x_{u,v}$ , using operational window  $W$  containing  $n = w^2$  samples. Pixels in  $W$  are denoted  $x_1 \dots, x_n$ , and  $x_1 = x_{u,v}$  is the center pixel of  $W$  (Fig. 2).

### 2.1 Noise detection

The *noise detection* phase is composed of the following steps:

- I. **Evaluation of pixels impulsiveness** begins with computation of dissimilarity measure  $d(x_1, x_j)$  between the central pixel and every other pixel contained in

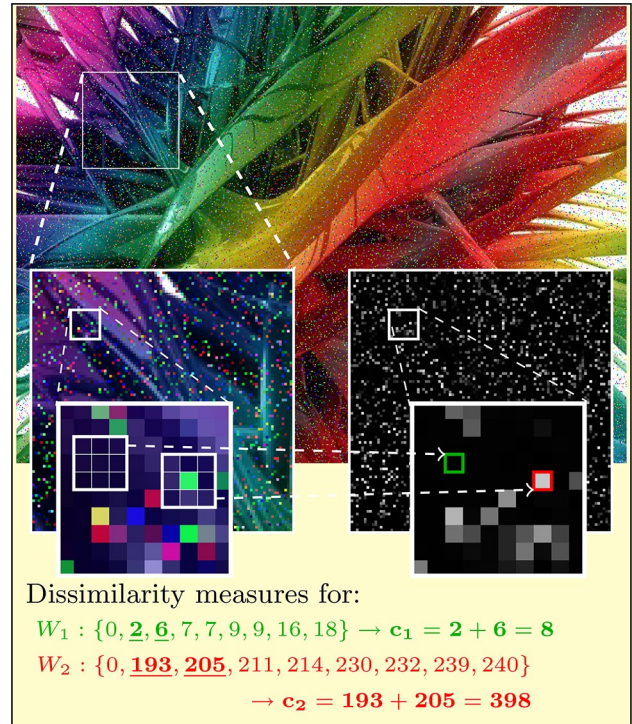


Fig. 3 Computation of pixel raw impulsiveness  $c$  (for two exemplary operating windows:  $W_1$  and  $W_2$ )

$W$ , denoted as  $\delta_i$ . Originally, the Euclidean distance was used, but many other dissimilarity measures can be used instead [11]. For example, in [58], the authors show that the use of Chebyshev distance ( $L_\infty$ ) improves detection performance, because this way algorithm is then more sensitive to outliers occurring on individual channels. Next, distances  $\delta_i$  (excluding  $\delta_1$  which is equal to 0) are sorted in ascending order:  $\delta_2, \dots, \delta_n \rightarrow \delta_{(1)}, \dots, \delta_{(n-1)}$ , and the trimmed sum of  $\alpha = 2$  smallest distances is computed for pixel  $x_{u,v}$ :

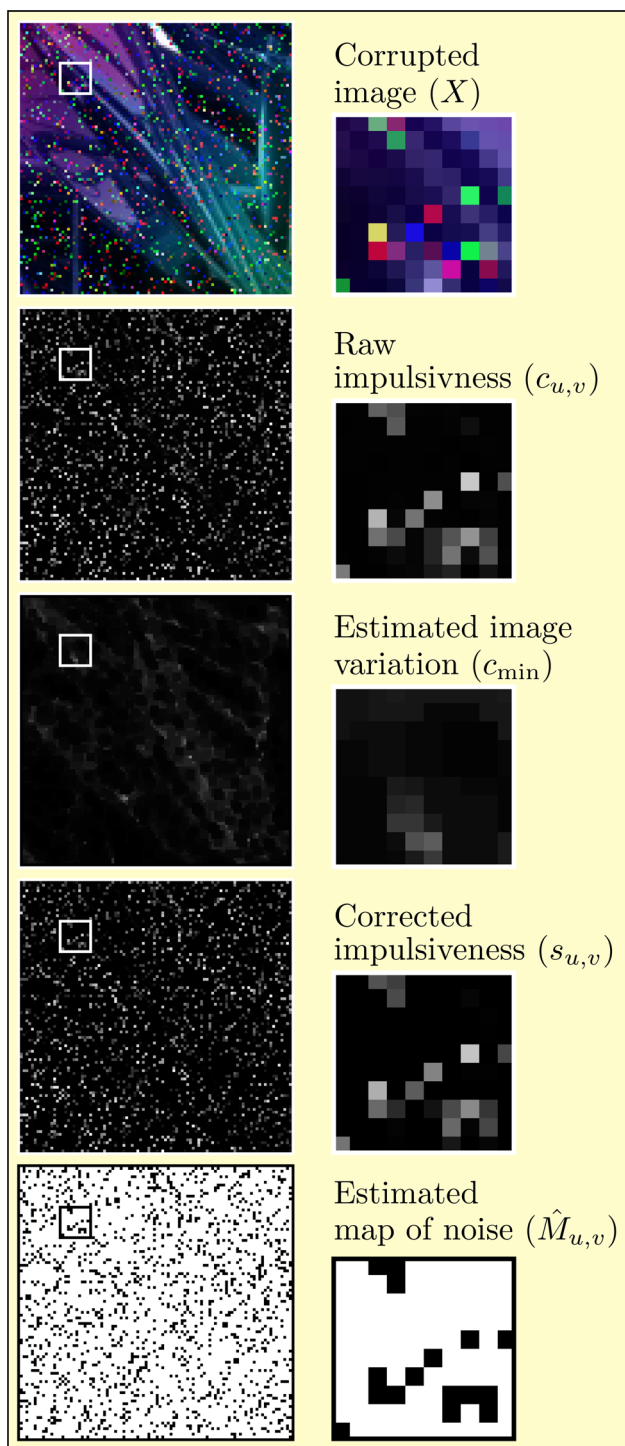
$$c_{u,v} = \sum_{r=1}^{\alpha} \delta_{(r)}. \tag{7}$$

The  $c_{u,v}$  can be interpreted as raw impulsiveness of the pixel (Fig. 3).

- II. **Adaptation to local image variation** is performed. For every  $c_{u,v}$ , a window  $W_c$ , containing  $n$  values  $c_i$ , is taken, so that  $c_1 = c_{u,v}$  is in the center of that window. The final corrected measure of corrected pixel impulsiveness (Fig. 4) assigned to pixel  $x_{u,v}$  is obtained:

$$s_{u,v} = c_{u,v} - c_{\min}, \tag{8}$$

where  $c_{\min} = \min\{c \in W_c\}$ . This correction normalizes impulsiveness based on the local image variation. In homogeneous image regions  $c_{\min}$  is close to 0



**Fig. 4** Transition from raw impulsiveness to estimated map of noise (using estimate of the image structure for impulsiveness correction)

and it rises together with variation in local neighborhood. As a result, the pixels of high raw impulsiveness in harsh regions of the image are less likely to be classified as noisy pixels of the same raw impulsiveness in smooth areas.

III. **Noise map acquisition** finalizes the *noise detection* phase, during which the estimated noise map  $\hat{M}$  is obtained. It is achieved by the comparison of  $s_{u,v}$  to the threshold  $t$ , provided by the user, for every pixel in the image  $X$  as follows:

$$\hat{m}_{u,v} = \begin{cases} 0 & \text{if } s_{u,v} > t, \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

The labeling of noisy pixels as 0 in  $\hat{M}$  is needed for the subsequent *pixel replacement* phase.

### 2.2 Pixel replacement

In the *pixel replacement* phase, the output image is obtained according to the following rule:

$$\hat{x}_{u,v} = \begin{cases} \text{AMF}(W) & \text{if } \hat{m}_{u,v} = 0, \\ x_{u,v} & \text{otherwise,} \end{cases} \tag{10}$$

where  $\text{AMF}(W)$  is the arithmetic mean computed only on members of  $W$ , which were designated as noise-free. In rare occasions (occurring for very high noise densities), when there is no noise-free pixels in  $W$ , the output is determined using the VMF scheme.

### 3 Self-tuning

As it has been shown in Fig. 1, there are three inputs for the algorithm: processed image ( $X$ ), threshold ( $t$ ), and operational window size ( $w$ ). As long as  $w$  is intuitive parameter to adjust, the proper choice of  $t$  may be a difficult one. It was shown in [54, 58] that optimal choice of  $t$  is dependent on impulsive noise density ( $\rho$ ), which is mostly unknown in real-case scenarios, so the operator is forced to experimental search of adequate value of  $t$ .

To free the user from manual adjusting of this parameter, a self-tuning modification is introduced. The main concept of this improvement is to use the estimated noise map  $\hat{M}$  (obtained during *noise detection* phase) to compute the estimated noise density  $\hat{\rho}$ . Combining  $\hat{\rho}$  with proper tuning characteristics like provided in [58] enables to adjust the  $t$  value, which can be used to obtain more accurate noise map.

#### 3.1 Algorithm

Based on the aforementioned idea, the self-tuning modification is introduced (Fig. 5). Before execution of the algorithm, the input  $t$  is set to initial value  $t_1 = 60$  (recommended value for  $t$  in FASTAMF using Chebyshev

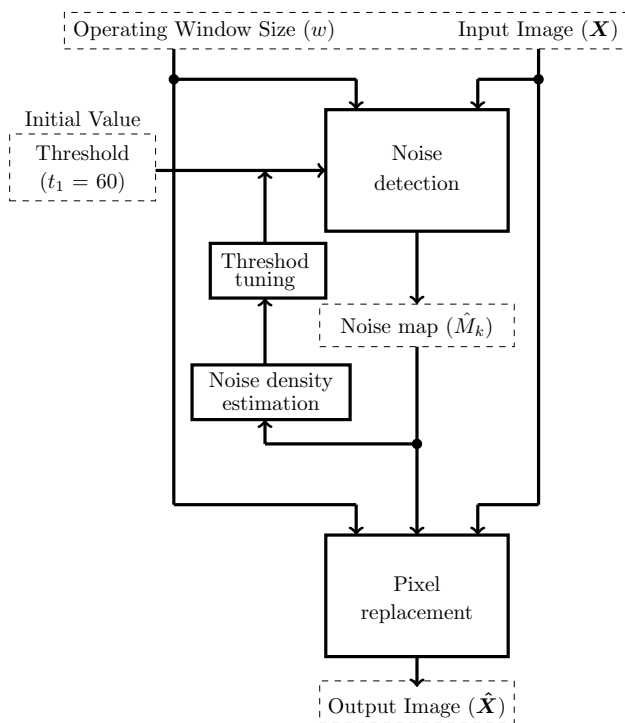


Fig. 5 Block diagram of self-tuning FASTAMF

distance, obtained experimentally). Then, the *noise detection* is performed until corrected impulsiveness measure is obtained for every pixel in the input image  $X$ .

Next, the recursive procedure of automatic  $t$  adjustment is performed by the following steps:

- The estimated map of noise for the current iteration  $\hat{M}_k$  is obtained by (9) using  $t_k$ .
- The estimated noise density for the current iteration  $\hat{\rho}_k$  is evaluated:  $\hat{\rho}_k = n_k/\theta$ , where  $n_k$  is the number of pixels designated as corrupted (in  $k$ th iteration) and  $\theta$  is the number of pixels in image  $X$ .
- $t_{k+1}$  value is interpolated (simple linear interpolation between two closest values) using tuning tables (see Table 4), which were obtained using procedure presented in Sect. 3.2.

Steps (a)–(c) are repeated in a loop until desired convergence  $|t_{k+1} - t_k| < \epsilon$  is achieved, where  $\epsilon$  regulates the convergence. In all experiments presented in this paper,  $\epsilon = 1$  was used.

Final estimated map of noise, denoted as  $\hat{M}$ , is then taken as an input to the *pixel replacement* phase. It is important that only the final step of the entire *noise detection* phase (9) has to be recursively repeated, so the increase in computational cost is not significant.

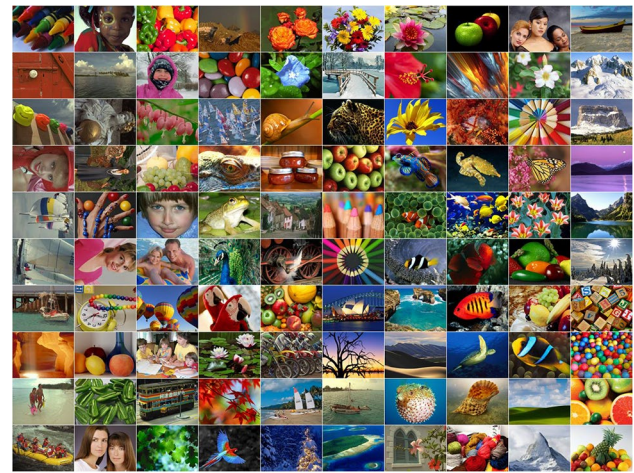


Fig. 6 Training image set

Although it might be tempting to design a similar solution for the adaptive tuning of operation window size  $w$ , it is pointless to do so due to the following reasons:

- It is very intuitive to choose  $w$  value, and using windows larger than  $3 \times 3$  is reasonable for high noise intensities only ( $\rho > 50\%$ ).
- Window size  $w$  has a critical impact on computational cost of the algorithm, so its automatic on-the-fly tuning certainly makes its execution time extremely unpredictable.
- Alteration of the  $w$  during algorithm's execution requires repetition of the entire *noise detection* phase, which is very costly from computational point of view. Therefore, such tuning algorithm would be inapplicable for real-time image processing tasks.
- Preliminary tests (omitted in the paper) revealed that  $w$  has stronger impact on *noise detection* phase performance than on *pixel replacement* phase. Therefore, partial solution, assuming the use of altered on-the-fly  $w$  for *pixel replacement* phase only, resulted in lack of restored image quality improvement.

### 3.2 Tuning tables

The core of the self-adjusting threshold  $t$  modification is the tuning Table 4 which provides the  $t$  values for interpolation step (c). Originally, this table was proposed for CTRI and CIRI noise models in [58]; however, in this paper, more thorough experiments were performed, to obtain more general insight into the problem.

The set of 100 color images has been taken as the training set (Fig. 6) [59]. Each of those images was artificially contaminated with CIRI, CCRI (correlation coefficient set to 0.5) and CTRI model for noise densities

**Table 1** Optimal and recommended  $t$  values for CIRI noise model

$\rho$ (%)	Mean (standard deviation) of optimal $t$ obtained using					$t$
	ACC	PSNR	MAE	NCD ( $10^{-4}$ )	FSIMc	
0.1	111.82 (29.95)	114.57 (38.26)	117.88 (36.22)	100.57 (28.85)	110.63 (36.86)	111
1	79.73 (21.57)	82.10 (24.91)	85.61 (23.58)	68.75 (19.59)	78.13 (24.55)	79
5	56.87 (15.77)	63.92 (18.25)	65.65 (17.60)	50.04 (13.51)	62.29 (17.57)	60
10	47.73 (13.11)	56.99 (15.44)	58.75 (15.19)	43.11 (11.58)	56.10 (14.85)	53
15	42.51 (11.07)	53.34 (14.33)	54.29 (14.45)	39.26 (9.60)	52.59 (13.81)	48
20	39.22 (10.06)	49.90 (12.83)	51.39 (12.75)	36.70 (9.24)	49.88 (12.38)	45
25	36.43 (8.90)	47.44 (12.28)	49.00 (12.07)	34.58 (8.18)	48.21 (11.97)	43
30	34.52 (8.21)	45.50 (11.40)	46.97 (11.11)	32.77 (7.38)	46.20 (11.95)	41
35	32.70 (7.25)	43.12 (10.30)	44.98 (10.60)	31.10 (6.75)	44.21 (11.18)	39
40	31.04 (6.79)	41.10 (9.73)	43.17 (9.72)	29.56 (6.28)	42.27 (10.67)	37
45	29.73 (6.09)	38.28 (9.39)	41.32 (9.44)	27.63 (5.91)	40.02 (10.72)	35
50	28.43 (5.40)	35.44 (8.64)	38.74 (9.00)	25.38 (5.46)	37.07 (10.38)	33
55	26.99 (4.72)	31.83 (8.40)	36.08 (8.46)	22.69 (5.25)	33.87 (10.56)	30
60	25.24 (4.00)	27.65 (8.38)	32.52 (8.73)	19.41 (4.80)	29.74 (10.65)	27
65	22.60 (3.31)	22.25 (7.77)	27.78 (8.45)	15.66 (4.54)	25.67 (10.53)	23
70	19.10 (2.63)	17.56 (7.03)	23.12 (7.87)	12.19 (4.06)	21.44 (10.57)	19
75	14.45 (2.09)	12.90 (6.24)	18.42 (7.29)	9.55 (3.45)	17.06 (9.89)	14
80	9.24 (1.57)	9.07 (4.68)	14.25 (6.44)	7.49 (2.78)	13.54 (9.33)	11

**Table 2** Optimal and recommended  $t$  values for CCRI noise model

$\rho$ (%)	Mean (standard deviation) of optimal $t$ obtained using					$t$
	ACC	PSNR	MAE	NCD ( $10^{-4}$ )	FSIMc	
0.1	111.19 (29.58)	115.77 (38.76)	119.12 (36.95)	100.27 (29.95)	109.24 (37.77)	111
1	78.33 (21.76)	82.59 (24.95)	84.97 (23.81)	66.86 (18.62)	77.47 (24.69)	78
5	54.62 (15.90)	63.08 (18.69)	65.42 (18.34)	47.88 (13.56)	61.22 (17.73)	58
10	44.80 (12.70)	55.26 (15.60)	57.06 (15.67)	40.81 (10.95)	54.54 (15.19)	50
15	39.58 (10.96)	51.05 (14.56)	53.09 (14.58)	36.98 (9.78)	50.92 (14.01)	46
20	36.23 (9.75)	47.86 (13.38)	49.94 (13.35)	34.34 (9.02)	48.69 (13.05)	43
25	33.33 (8.74)	45.53 (12.14)	47.63 (12.42)	32.08 (7.93)	45.96 (12.46)	41
30	31.07 (7.69)	42.49 (11.53)	45.30 (11.79)	30.04 (6.93)	43.65 (11.78)	39
35	29.14 (6.82)	40.28 (10.63)	43.13 (11.12)	28.62 (6.63)	41.78 (11.58)	37
40	27.56 (6.25)	38.17 (10.30)	41.49 (10.69)	26.83 (6.23)	39.54 (11.20)	35
45	26.04 (5.65)	35.96 (9.97)	39.58 (9.94)	25.18 (5.56)	37.84 (10.59)	33
50	24.56 (5.06)	33.15 (9.10)	37.06 (9.46)	23.01 (5.18)	35.18 (10.96)	31
55	23.14 (4.50)	30.12 (8.89)	34.97 (8.97)	20.61 (4.88)	32.80 (10.61)	28
60	21.56 (3.62)	27.18 (8.50)	32.18 (8.75)	18.17 (4.69)	29.91 (10.91)	26
65	19.30 (3.04)	23.34 (8.12)	29.12 (8.74)	15.72 (4.65)	26.41 (10.89)	23
70	16.41 (2.50)	19.31 (7.34)	25.56 (8.28)	12.86 (4.17)	23.41 (10.74)	20
75	12.57 (1.88)	15.43 (6.83)	22.16 (8.30)	10.53 (3.71)	20.33 (10.98)	16
80	8.27 (1.35)	11.77 (5.78)	18.78 (7.85)	8.39 (3.16)	17.39 (10.64)	13

$\rho \in \{0.1, 1, 5, 10, 15, \dots, 80\}$ . Finally, for each contaminated image, the optimization has been performed to find optimal value of  $t$  for which ACC, PSNR, and FSIMc are maximal and MAE and NCD is minimal.

The mean values (and standard deviations) of optimal  $t$ , computed upon entire set of training images for a chosen performance measure, noise model, and noise density are presented in Tables 1, 2, and 3. The final proposition of general tuning values obtained as an average of results from all experiments is shown in Table 4.

**Table 3** Optimal and recommended  $t$  values for CTRI noise model

$\rho$ (%)	Mean (standard deviation) of optimal $t$ obtained using					$t$
	ACC	PSNR	MAE	NCD ( $10^{-4}$ )	FSIMc	
0.1	114.48 (29.20)	115.07 (34.46)	115.96 (32.98)	104.66 (26.34)	110.09 (33.29)	112
1	87.78 (19.69)	85.11 (22.69)	87.03 (21.39)	76.35 (18.30)	81.69 (22.64)	84
5	68.94 (14.99)	68.53 (16.50)	69.89 (15.53)	60.62 (13.01)	66.30 (15.47)	67
10	61.22 (12.57)	61.71 (13.63)	62.60 (12.98)	54.39 (11.36)	60.90 (13.85)	60
15	57.18 (11.21)	57.87 (12.01)	58.95 (11.70)	50.66 (9.61)	56.67 (11.66)	56
20	54.24 (9.93)	54.38 (10.23)	55.74 (10.16)	47.90 (8.95)	53.69 (10.73)	53
25	52.10 (8.88)	51.83 (9.15)	53.48 (9.37)	46.03 (7.43)	51.35 (9.87)	51
30	50.58 (8.25)	49.74 (8.36)	51.38 (8.25)	44.13 (7.23)	49.02 (9.20)	49
35	49.49 (7.27)	46.84 (8.28)	49.05 (8.06)	42.01 (6.66)	46.63 (9.23)	47
40	48.58 (6.61)	43.01 (7.56)	46.00 (7.81)	39.15 (6.30)	43.14 (9.36)	44
45	47.91 (5.35)	37.86 (8.48)	42.05 (8.13)	34.57 (6.14)	38.01 (10.30)	40
50	46.89 (4.81)	30.52 (8.83)	36.15 (8.48)	28.72 (6.53)	31.71 (10.34)	35
55	44.54 (4.21)	22.62 (7.83)	28.61 (8.59)	21.78 (5.89)	24.26 (9.87)	28
60	39.82 (4.78)	15.16 (5.74)	20.22 (7.42)	15.37 (4.62)	17.63 (8.46)	22
65	31.72 (4.73)	9.81 (3.81)	13.43 (5.11)	10.70 (3.25)	12.27 (6.01)	16
70	22.66 (4.48)	6.29 (2.16)	8.93 (3.17)	7.59 (2.10)	8.36 (4.51)	11
75	14.52 (3.14)	4.47 (1.29)	6.42 (1.92)	5.89 (1.42)	5.73 (3.18)	7
80	8.20 (1.83)	3.46 (0.78)	4.96 (1.20)	4.65 (1.03)	3.00 (2.64)	5

**Table 4** Tuning values for threshold  $t$ 

$\rho$ (%)	0.1	1	5	10	20	25	30	35	40	45	50	55	60	65	70	75	80
$t$	111	80	61	54	50	47	45	43	41	38	36	33	28	25	20	16	12

**Fig. 7** Validation image set. The images are numbered from 1 (top left) to 10 (bottom right)

## 4 Noise suppression performance

Our aim was to provide the most objective noise suppression performance test; therefore, a new set of ten images was taken as validation set (Fig. 7) [60]. In addition, all of those validation images were contaminated with CPRI noise ( $p_R = p_G = p_B = p_A = 0.25$ ), which has not been used for obtaining the tuning Table 4. This way, we provided an independent test input, further minimizing the possibility that the tuning values are optimized for particular image set or noise model.

### 4.1 FASTAMF compared with state-of-the-art algorithms

While original FASTAMF algorithm [54] was operating on Euclidean distance, the new version uses its Chebyshev counterpart. Therefore, a new comparison to state-of-the-art filters is required. This time, we decided to restrict the state-of-the-art algorithm base to four filters, which were found to be the most competitive using the recommended parameter settings:<sup>1</sup>

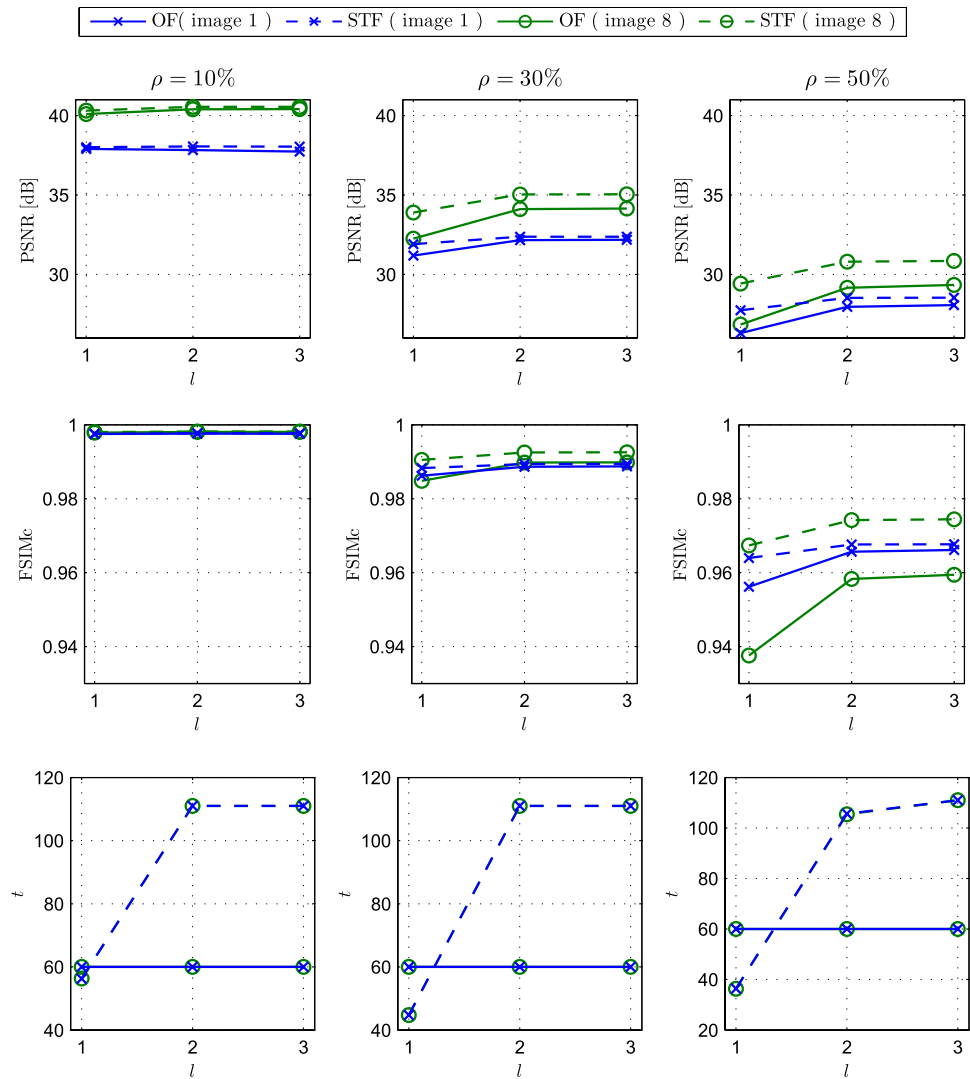
- FASTAMF with recommended  $t = 60$ .
- ACWVMF [61] with  $\lambda = 2$  and  $Tol = 80$ .
- FAPGF [42] with  $d = 0.1$  and  $\gamma = 0.8$ .
- FFNRF [48] with  $K = 1024$  and  $\alpha = 3.5$ .
- FPGF [41] with  $m = 3$  and  $d = 45$ .

For all tested algorithms, the operating window size was set to  $w = 3$  and the comparison was performed for noise densities  $\rho \in \{10, 20, \dots, 50\%\}$ . For each corrupted image from validation set and for each tested algorithm, the noise

<sup>1</sup> The notation of parameters used in the respective papers was adopted.



**Fig. 8** Algorithm’s performance in subsequent iterations for validation images 1 and 8 (for which STF performs better)



suppression was performed, and PSNR, MAE, NCD, and FSIMc measures were calculated. The results were grouped by each measure and noise density and compared using statistical tests.

For all results in the test group, the Friedman’s test [62] was performed. Two opposite hypotheses were taken under consideration:

- H0: There **is no** evidence that results for all algorithms are significantly heterogeneous.
- H1: There **is** evidence that results for all algorithms are significantly heterogeneous.

For each group for which H0 has been discarded in favor of the H1, the set of Friedman—Post hoc tests proposed by Nemenyi [62]—was performed comparing FASTAMF to each other algorithm. For those tests, the following hypotheses were stated:

- H2: There **is no** evidence that FASTAMF performs significantly better than the compared algorithm.
- H3: There **is** evidence that FASTAMF performs significantly better than compared algorithm.

The results of above tests are summarized in Table 5 (Friedman’s tests) and in Table 6 (Post hoc tests). All emboldened values in the tables do not support the superiority of the FASTAMF (as those are in minority). In case of PSNR and FSIMc measures, the better value is the higher one, so the higher mean rank values support the superiority of particular algorithm. The opposite situation is for MAE and NCD measures which are better if smaller.

The following conclusions can be drawn:

- The results obtained from all algorithms (represented by quality measures) were always heterogeneous (H0 was discarded in favor of H1 in every case). In addition,  $p$  in

**Table 5** Friedman’s test results—FASTAMF compared with state-of-the-art filters

$\rho$ (%)	Mean ranks for					H1
	FASTAMF	ACWVMF	FAPGF	FFNRF	FPGF	
Friedman’s test for PSNR						
10	5.0	3.8	2.9	1.5	1.8	$p < 0.001$
20	4.9	3.0	3.0	1.4	2.7	$p < 0.001$
30	4.9	2.0	3.1	1.7	3.3	$p < 0.001$
40	5.0	1.4	3.7	2.0	2.9	$p < 0.001$
50	4.9	1.2	4.0	2.0	2.9	$p < 0.001$
Friedman’s test for MAE						
10	1.1	2.0	3.4	3.7	4.8	$p < 0.001$
20	1.0	2.2	3.7	3.5	4.6	$p < 0.001$
30	1.0	3.5	3.6	2.9	4.0	$p < 0.001$
40	1.0	4.2	3.2	2.8	3.8	$p < 0.001$
50	1.0	4.4	2.8	2.7	4.1	$p < 0.001$
Friedman’s test for NCD						
10	1.1	2.1	3.5	4.2	4.1	$p < 0.001$
20	1.0	2.8	3.6	4.3	3.3	$p < 0.001$
30	1.0	4.2	3.1	3.8	2.9	$p < 0.001$
40	1.0	4.7	2.9	3.6	2.8	$p < 0.001$
50	1.0	4.9	2.8	3.6	2.7	$p < 0.001$
Friedman’s test for FSIMc						
10	5.0	3.3	3.5	1.5	1.7	$p < 0.001$
20	4.9	2.7	3.5	1.6	2.3	$p < 0.001$
30	4.9	1.7	3.3	2.3	2.8	$p < 0.001$
40	5.0	1.6	3.3	2.5	2.6	$p < 0.001$
50	4.9	1.1	3.4	3.1	2.5	$p < 0.001$

each Friedman’s test was very low, so the differences in results are unquestionably significant.

- For every measure and noise density, the best mean ranks were observed for FASTAMF algorithm, which means that it was the best or almost the best for every tested image.
- Only very few of Post hoc tests resulted in favor of H2 hypothesis. For those rare cases, we can state that FASTAMF is not significantly better than the compared algorithm. For each other case, however, it is the best performing algorithm among tested.
- For low-noise densities, the ACWVMF tends to be a competitive choice for FASTAMF, while, for higher noise contamination ratios, the FAPGF provides the most similar results.

## 4.2 Self-tuning FASTAMF against original FASTAMF

The main goal of self-tuning feature is to free the user from experimental threshold selection, which is optimal for a given noise density. Therefore, self-tuning FASTAMF (further denoted as STF) is compared to the original FASTAMF (further denoted as OF from Original Filter) with recommended  $t = 60$ . This time, only two algorithms were

compared, so Wilcoxon’s test [62] was used (not every sample has normal distribution, so t test cannot be performed). The following hypotheses were formulated:

- H0: There **is not** enough evidence that STF provides significantly better results.
- H1: There **is** enough evidence that STF provides significantly better results.

The results are presented in Table 7. This time, all emboldened values support the superiority of STF algorithm). The results can be summarized as follows:

- Larger Positive Sums of Ranks for PSNR and FSIMc quality measures indicate that STF algorithm performs better (values of measure are more frequently higher). Such outcome can be observed for  $\rho \geq 20\%$ , for both measures.
- Smaller Positive Sums of Ranks for NCD and MAE measures indicate that STF algorithm performs better (values of measure are more frequently lower). Such outcome can be observed for MAE when  $\rho \geq 40\%$  and for all results evaluated with NCD.
- For  $\rho \geq 40\%$  the STF algorithm performs significantly better than OF from NCD and PSNR point of view.

**Table 6** Post hoc test results—FASTAMF compared with state-of-the-art filters (emboldened results speak against FASTAMF superiority)

$\rho$ (%)	Hypothesis ( $p$ ) for			
	ACWVMF	FAPGF	FFNRF	FPGF
Post hoc tests for PSNR				
10	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
20	H3 ( $< 0.05$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
30	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )
40	H3 ( $< 0.01$ )	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
50	H3 ( $< 0.01$ )	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
Post hoc tests for MAE				
10	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
20	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
30	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )
40	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )
50	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )
Post hoc tests for NCD				
10	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
20	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
30	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )
40	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )
50	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )
Post hoc tests for FSIMc				
10	H3 ( $< 0.05$ )	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
20	H3 ( $< 0.01$ )	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
30	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
40	H3 ( $< 0.01$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )	H3 ( $< 0.01$ )
50	H3 ( $< 0.01$ )	<b>H2</b> ( $> 0.05$ )	H3 ( $< 0.05$ )	H3 ( $< 0.01$ )

The results obtained are very satisfactory. It is obvious that ST modification is not significantly better for lower  $\rho$  values due to the recommended fixed  $t$  being suitable for such scenarios. In addition, while  $\rho$  becomes higher, the better performance of STF becomes more noticeable, as this algorithm automatically adjusts its optimal threshold value.

### 4.3 Multi-run and visual comparison

One of the common approaches to achieve good noise suppression performance is to repeat the processing of the noisy picture several times, using output image as an input for next algorithm's execution. This way noisy pixels omitted during first filtering may be detected and restored during subsequent runs. However, this approach may lead to stronger degradation of image details, especially if the algorithm has adaptive features.

The noise suppression scheme (further referenced as multi-run or MR) was performed for three iterations ( $l = 1, 2, 3$ ) upon all the validating images for noise densities

$\rho \in \{10, 30, 50\}$ , and the four representative images were selected for detailed comparison (validation images 1, 7, 8, and 9).

To provide fair comparison, two images for which OF algorithm achieved a better performance in the MR test (validation images 7 and 9) were opposed to two images for which ST provided better results (validation images 1 and 8). The OF scheme was applied for three iterations with the same recommended  $t = 60$ , while STF algorithm calculated threshold value automatically in each iteration.

The efficiency of both algorithms in terms of PSNR and FSIMc measures is presented in Table 8. In addition, the visual comparison of performance for both filtering schemes for  $l = 1, l = 3$  and  $\rho = 30\%$  is depicted in Figs. 10, 11, 12, and 13.

As can be seen, the STF algorithm performs better for “easier” tasks (validation images 1 and 8)—which are meager in detail and have large homogeneous regions (Figs. 10 and 12). The threshold value  $t$  is well adjusted in the first execution (Fig. 8), and then, it is set automatically to higher value in subsequent runs. This is caused by low estimated noise density after first noise suppression and is beneficial for the purpose of detail preservation. The STF algorithm does not try to repair less explicit outliers, which might be the image details.

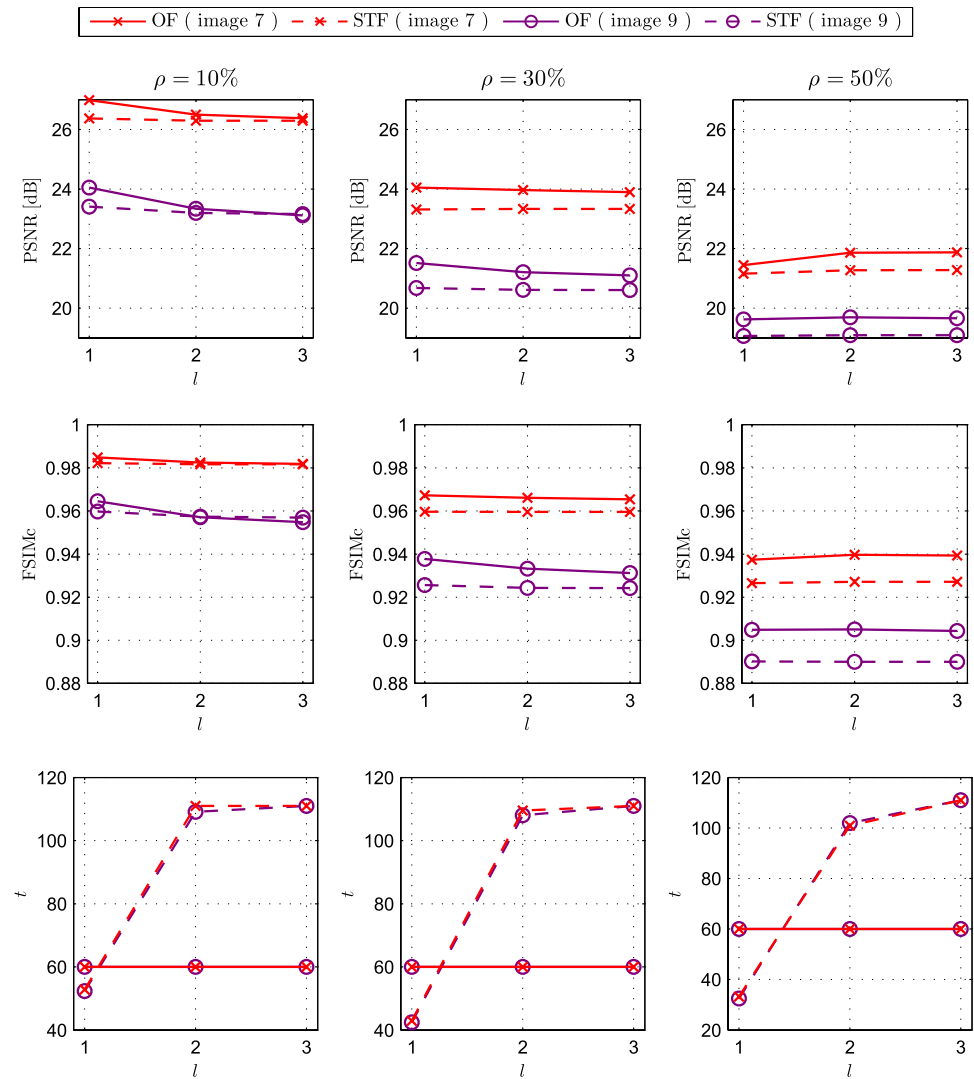
For harder tasks, however (validation images 7 and 9)—images rich in small details (Figs. 11 and 13)—a large value of ST might be too high to enable the restoration of omitted noisy pixels (Fig. 9). The OF algorithm with fixed  $t$  shows higher efficiency, while it tends to restore pixels with the same impulsiveness.

We can observe that although the self-tuning feature of the algorithm is very convenient and may achieve a better statistical performance, especially if the noise density is unknown or non-stationary, it may achieve slightly inferior efficiency than fixed  $t$  value version for more complicated images.

A more detailed analysis of zoomed regions on images 1 and 8 (Figs. 10, 11, 12, 13) shows that:

- the OF removes less noisy pixels for  $l = 1$  than its STF counterpart (Fig. 10c, d), due to higher value of  $t$ . If the reason behind those leftovers is high variance of the local area, those are mostly removed in subsequent iterations (Fig. 10e). If a too high  $t$  value caused this omission, those will not be restored, no matter how many iterations will be performed. In addition, fixed  $t$  value makes OF completely insensitive to less explicit noisy pixels, which is reflected in numerical (PSNR) and structural (FSIMc) measures;
- the STF scheme removes more noisy pixels in the first execution (Fig. 10d), strongly decreasing the local variance of the image. Consequently, it is easier to remove

**Fig. 9** Algorithm's performance in subsequent iterations for validation images 7 and 9 (for which OF performs better)



omitted noisy pixels in subsequent iterations (Fig. 10f), and there is also lower count of less explicit noisy pixels due to lower  $t$  in the first run;

- the STF scheme tends to remove more details (Fig. 11d, f) from the image (it is more blurry), than OF (Fig. 11c, e). As long as it is hard to be noticed visually without zoom, it clearly affects numerical (PSNR) and also structural (FSIMc) measures;
- undetected by OF noisy pixels during the first execution (Fig. 12c), may cause low-level distortions around them, which will not be repaired in subsequent iterations (Fig. 12e). Such phenomenon is far less noticeable if STF scheme is used (Fig. 12d, f).
- STF scheme tends to remove more details in the most difficult cases (Fig. 13d, f), which is reflected mostly by the PSNR measure.

It has to be pointed out that images 1 and 8 contain large homogeneous regions which makes the adjustment of  $t$  easier. In contrast, the high local variance of regions occurring in images 7 and 9 makes  $t$  tuning harder. As long as very large number of distinct images was used as training set for obtaining the tuning Table 4, the local variance of the image has not been taken into account, nor it is measured in any way during  $t$  adjustment. Such approach was considered and tested in the early stages of STF development, but it was very computationally expensive, so not applicable for real-time implementations.

The visual comparison shows that STF algorithm seems to always achieve a better noise suppression efficiency (less explicit leftovers can be noticed). Therefore, lower PSNR values might be caused by very subtle differences, which can be detected on the numerical level only.

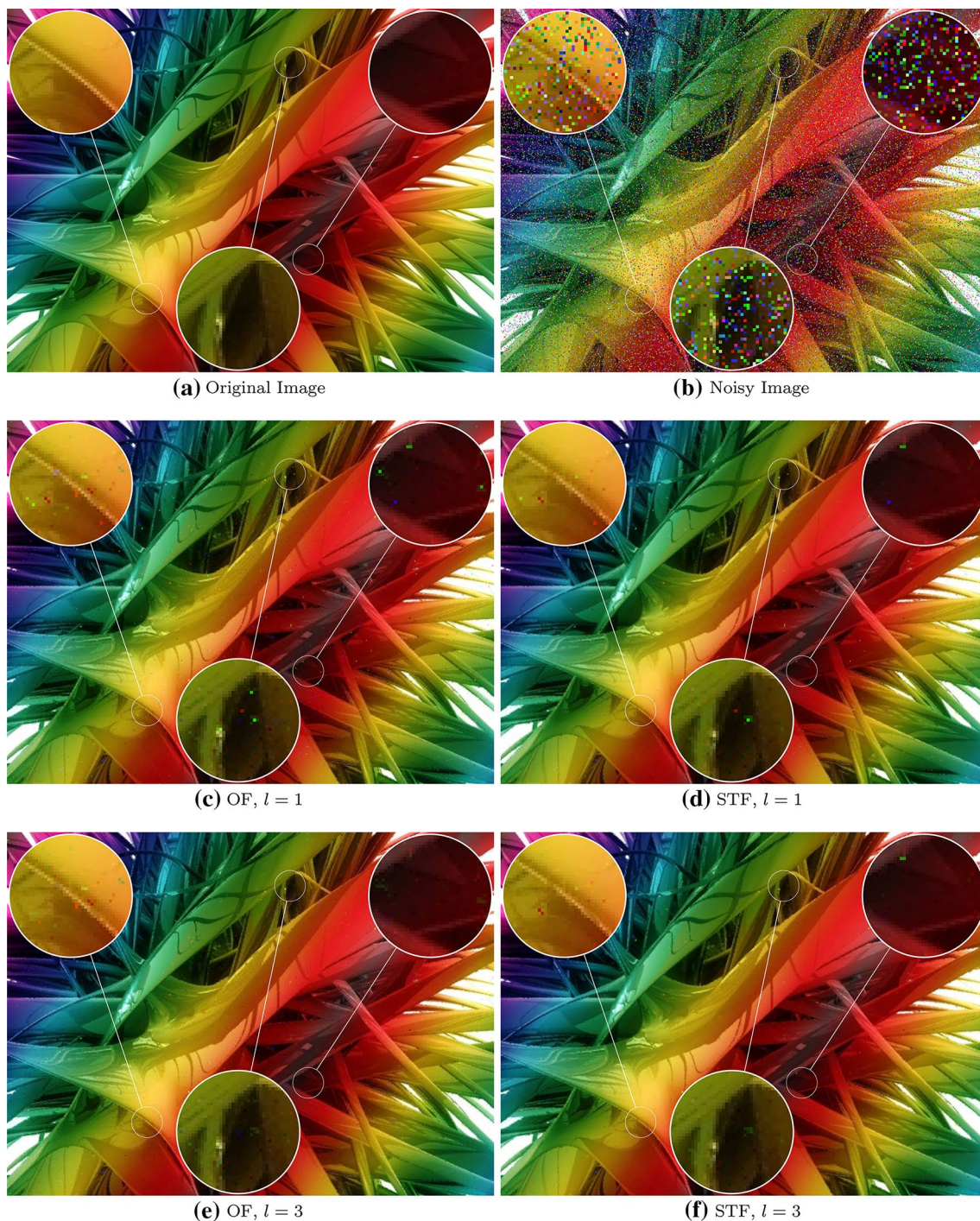


Fig. 10 Visual comparison of OF and STF performance for  $\rho = 30\%$  and validation image 1

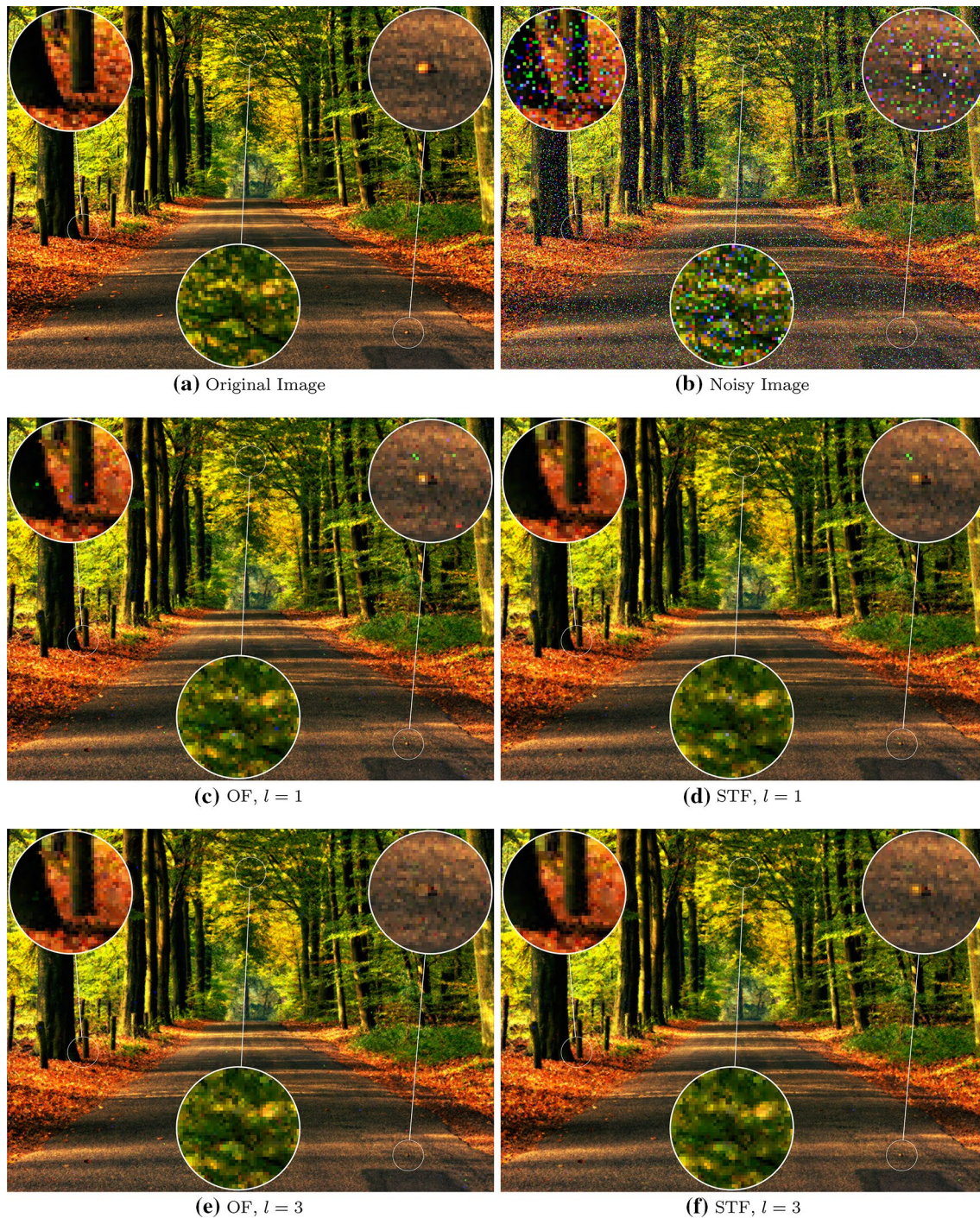
## 5 Efficiency

### 5.1 Computational complexity

A detailed analysis of computational complexity of algorithm is presented in [54]; therefore, in this paper, it has been performed for ST modification only. Self-tuning begins

after the computation of corrected impulsiveness and in each iteration  $k$  requires:

1. Estimation of map of noise  $\hat{M}_k$  which needs  $\mu \times \nu$  comparisons (COMPS). This step has linear complexity.



**Fig. 11** Visual comparison of OF and STF performance for  $\rho = 30\%$  and validation image 7

2. Estimation of noise density  $\hat{\rho}_k$  for which  $\mu \times \nu$  additions (ADDS) and one division (DIVS) are necessary. This step also has linear complexity.
3. Linear interpolation of  $t$ :

$$t_k = \frac{t_A - t_B}{\rho_A - \rho_B} \hat{\rho}_k + \left( t_A - \frac{t_A - t_B}{\rho_A - \rho_B} \rho_A \right), \quad (11)$$

where  $A$  and  $B$  are the nearest indicies of values in Table 4 for which  $\rho_A \leq \hat{\rho}_k \leq \rho_B$ . It demands 5 subtractions (SUBS), 2 divisions (DIVS), 2 multiplications (MULTS), and 1 addition (ADDS) and up to 18 COMPS (required for determination for  $A$  and  $B$ ). This step is not image size-dependent, so it can be treated as step with constant computational complexity.

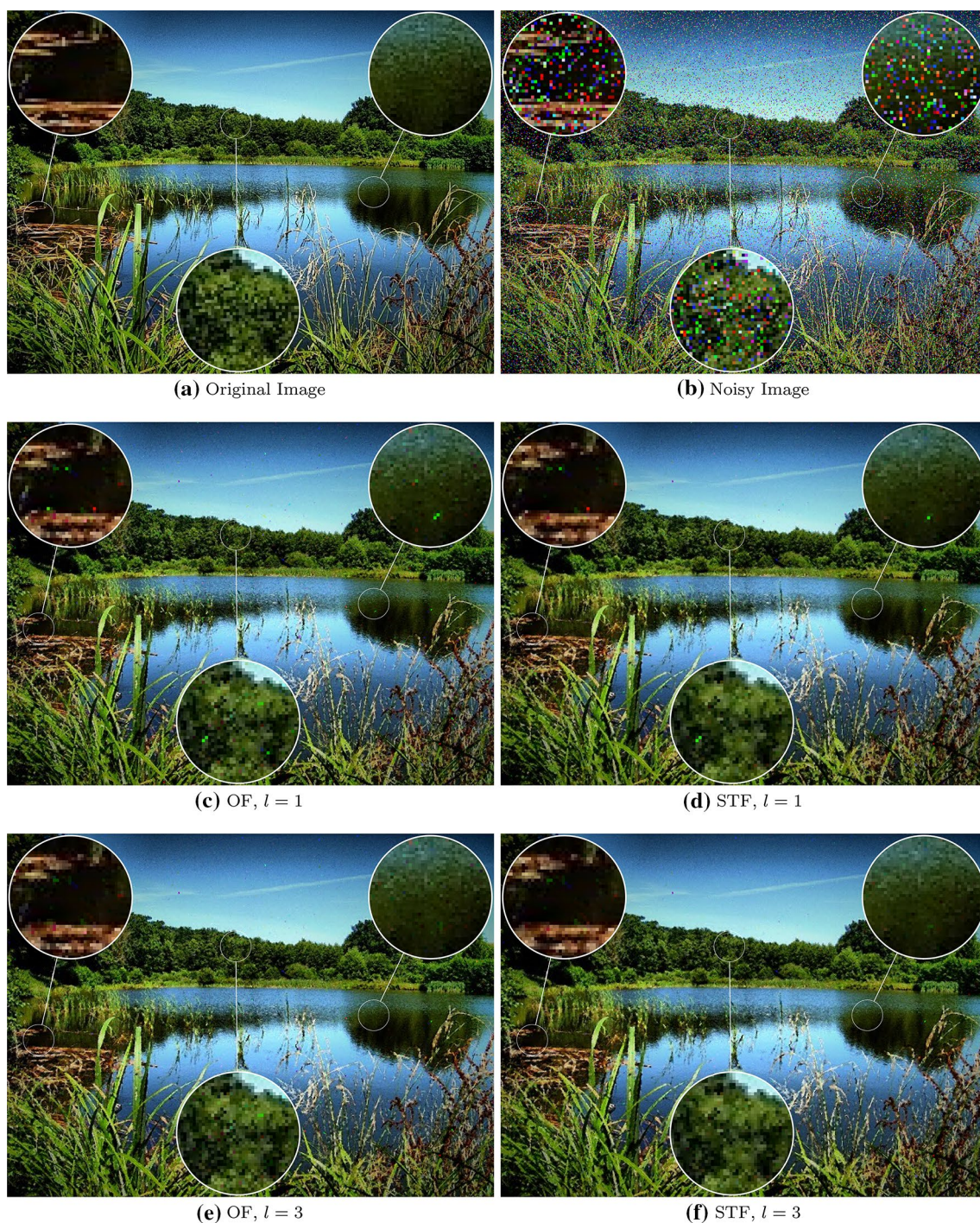


**Fig. 12** Visual comparison of OF and STF performance for  $\rho = 30\%$  and validation image 8

- Convergence condition check which also has constant computational complexity and requires one subtraction, and two comparisons.

The remaining issue is the number of iterations required to achieve desired convergence. For every image in training image set, noise models: CTRI, CIRI, and CCRI, and

noise densities  $\rho \in \{0.1, 1, 5, 10, 15, \dots, 80\%$  STF algorithm has been executed and the number of iterations (denoted as  $k_F$ ) needed to satisfy the convergence condition has been obtained. The results are presented in Table 9. It can be seen that the  $k_F$  is very stable and has almost deterministic value.



**Fig. 13** Visual comparison of OF and STF performance for  $\rho = 30\%$  and validation image 9

The computational cost of single iteration of ST modification is not very heavy and is linearly dependent on image size. The number of iteration required to archive final  $t$  values is fairly low and predictable, so this modification is a suitable addition to FASTAMF algorithm in terms of real-time image processing requirements.

## 5.2 Experimental comparison

The execution time and noise suppression efficiency of FASTAMF with ST modification has been compared to the original FASTAMF (with recommended  $t = 60$ ). In tests, all ten images' validation set (Fig. 7) contaminated with CPRI model and noise densities  $\rho \in \{10, 20, 30\%\}$  is used.



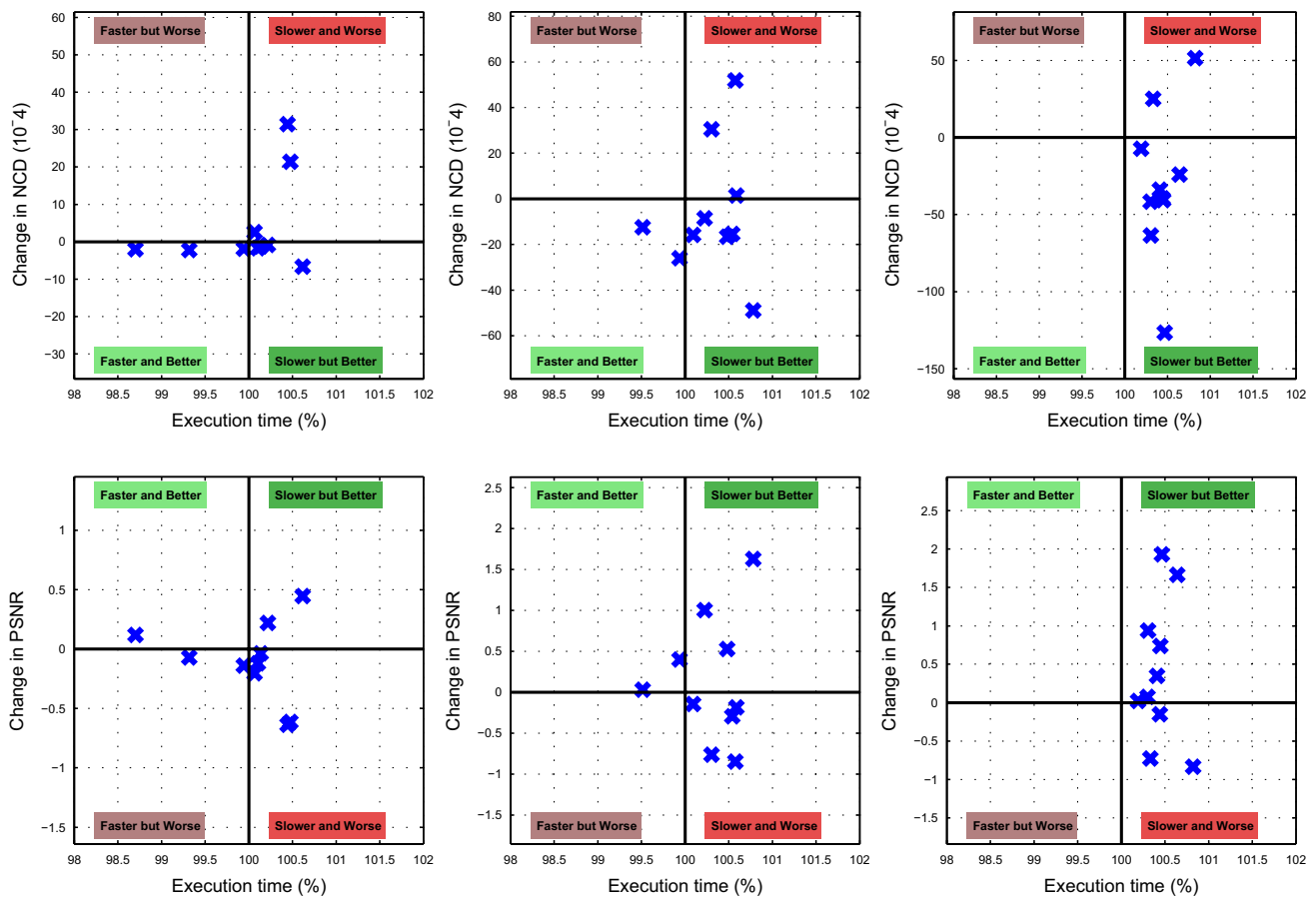


Fig. 14 Execution time and performance of ST version in comparison to the original FASTAMF algorithm

The noise suppression efficiency has been evaluated by PSNR, MAE, NCD, and FSIMc measures. Since the tuning tables were obtained as the trade-off between those measures, in Fig. 14, the most favorable (NCD) and most adverse (PSNR) outcomes of using ST modification were presented. On vertical axis, the difference in particular measure is presented, while, on horizontal one, the change of execution time (in percentages) is exhibited. The point (100,0) refers to all results obtained using original FASTAMF algorithm, and marked points represent results obtained for ST version using ten test images.

Interestingly, in individual cases for  $\rho = 10\%$ , the ST version might be even faster than original algorithm, because the threshold value is calculated to be higher than  $t = 60$ . As a consequence, fewer pixels are recognized as noisy, and noise suppression (AMF) has less work to do.

Also the major conclusion is that results for ST modification become better, along with increasing noise density.

## 6 Summary

The achieved denoising results are very satisfactory, since the reduction of the number of FASTAMF parameters and a better overall performance have been the main goal of this research. The new self-tuning FASTAMF, achieves slightly better or at least not worse overall performance than the original algorithm, yet it has no parameters which require experimental adjustment.

Also the computational cost of the self-tuning is not significantly higher, since it works after the most computationally expensive part of the algorithm (estimation of the pixel impulsiveness).

The major virtue of the self-tuning FASTAMF is its adaptability to noise density. As it can be useful for filtering of images contaminated by impulsive noise of unknown density, it might be even more advantageous for processing of video sequences distorted by noise with time-dependent parameters. The initial value of  $t$  (for self-tuning mechanism) can be carried out from frame to frame, in such

**Table 7** Wilcoxon’s tests for STF and OF comparison (emboldened results denotes a significant superiority of ST)

$\rho$ (%)	Sum of ranks		Hypothesis ( $p$ )
	Positive	Negative	
Wilcoxon’s test for PSNR			
10	18	37	H0 (> 0.05)
20	31	24	H0 (> 0.05)
30	39	16	H0 (> 0.05)
40	<b>46</b>	9	<b>H1</b> (< 0.05)
50	<b>51</b>	4	<b>H1</b> (< 0.05)
Wilcoxon’s test for MAE			
10	43	12	H0 (> 0.05)
20	36	19	H0 (> 0.05)
30	31	24	H0 (> 0.05)
40	25	30	H0 (> 0.05)
50	21	34	H0 (> 0.05)
Wilcoxon’s test for NCD			
10	26	29	H0 (> 0.05)
20	19	36	H0 (> 0.05)
30	11	44	H0 (> 0.05)
40	<b>2</b>	53	<b>H1</b> (< 0.05)
50	<b>0</b>	55	<b>H1</b> (< 0.05)
Wilcoxon’s test for FSIMc			
10	16	39	H0 (> 0.05)
20	28	27	H0 (> 0.05)
30	32	23	H0 (> 0.05)
40	39	16	H0 (> 0.05)
50	43	12	H0 (> 0.05)

**Table 8** Multi-run test (emboldened results are superior in each contamination rate)

$l$	Filter	Image 1			Image 7			Image 8			Image 9		
		$t$	PSNR	FSIMc	$t$	PSNR	FSIMc	$t$	PSNR	FSIMc	$t$	PSNR	FSIMc
$\rho=10\%$													
1	OF	60.0	37.90	0.9975	60.0	<b>26.99</b>	<b>0.9849</b>	60.0	40.09	0.9979	60.0	<b>24.05</b>	<b>0.9645</b>
2			37.83	0.9976		<b>26.50</b>	<b>0.9824</b>		40.40	0.9981		<b>23.34</b>	0.9571
3			37.73	0.9975		<b>26.37</b>	<b>0.9818</b>		40.41	0.9981		23.12	0.9548
1	STF	56.3	<b>38.01</b>	<b>0.9976</b>	52.8	26.37	0.9822	56.3	<b>40.31</b>	<b>0.9980</b>	52.3	23.41	0.9597
2		111.0	<b>38.06</b>	<b>0.9977</b>	111.0	26.29	0.9817	111.0	<b>40.55</b>	<b>0.9982</b>	109.1	23.20	<b>0.9573</b>
3		111.0	<b>38.04</b>	<b>0.9977</b>	111.0	26.29	0.9817	111.0	<b>40.55</b>	<b>0.9982</b>	111.0	<b>23.16</b>	<b>0.9569</b>
$\rho=30\%$													
1	OF	60.0	31.19	0.9862	60.0	<b>24.05</b>	<b>0.9672</b>	60.0	32.25	0.9848	60.0	<b>21.51</b>	<b>0.9378</b>
2			32.16	0.9886		<b>23.96</b>	<b>0.9661</b>		34.11	0.9898		<b>21.21</b>	<b>0.9332</b>
3			32.18	0.9887		<b>23.90</b>	<b>0.9654</b>		34.15	0.9898		<b>21.10</b>	<b>0.9312</b>
1	STF	44.7	<b>31.90</b>	<b>0.9883</b>	42.9	23.32	0.9596	44.7	<b>33.89</b>	<b>0.9905</b>	42.4	20.68	0.9256
2		111.0	<b>32.37</b>	<b>0.9894</b>	109.6	23.33	0.9595	111.0	<b>35.04</b>	<b>0.9925</b>	108.0	20.62	0.9243
3		111.0	<b>32.38</b>	<b>0.9894</b>	111.0	23.33	0.9596	111.0	<b>35.05</b>	<b>0.9926</b>	111.0	20.61	0.9242
$\rho=50\%$													
1	OF	60.0	26.31	0.9562	60.0	<b>21.45</b>	<b>0.9374</b>	60.0	26.84	0.9375	60.0	<b>19.63</b>	<b>0.9049</b>
2			27.96	0.9656		<b>21.86</b>	<b>0.9397</b>		29.16	0.9583		<b>19.70</b>	<b>0.9050</b>
3			28.07	0.9661		<b>21.87</b>	<b>0.9393</b>		29.34	0.9594		<b>19.66</b>	<b>0.9044</b>
1	STF	36.4	<b>27.74</b>	<b>0.9640</b>	33.2	21.16	0.9265	36.3	<b>29.42</b>	<b>0.9674</b>	32.5	19.07	0.8902
2		105.7	<b>28.52</b>	<b>0.9676</b>	100.9	21.27	0.9271	105.4	<b>30.80</b>	<b>0.9742</b>	101.9	19.09	0.8900
3		111.0	<b>28.54</b>	<b>0.9677</b>	111.0	21.28	0.9272	111.0	<b>30.84</b>	<b>0.9744</b>	111.0	19.10	0.8901

Bold values indicate to the best results acquired for particular image and noise ratio

**Table 9** The number of iterations ( $k_F$ ), required to achieve the ST convergence

$k_F$	$\rho$							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Min.	3	3	3	3	4	4	4	4
Med.	3	3	3	3	4	4	5	4
Max.	3	3	3	4	5	6	6	5

implementations, decreasing the number of potential iterations, required to achieve required convergence. The application of proposed filtering scheme to the video enhancement will be the subject of future work.

**Acknowledgements** This work was supported by a research Grant 2017/25/B/ ST6/02219 from the National Science Centre, Poland, and was also funded by the Silesian University of Technology, Poland (Grants BK 2018).

**Author contributions** LBM 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 digital image processing but also he explores field of multidimensional optimisation for nonlinear problems. BS 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. BS 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.

**OpenAccess** 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, New York (2000)
- Lukac, R., Smolka, B., Martin, K., Plataniotis, K., Venetsanopoulos, A.: Vector filtering for color imaging. *IEEE Signal Process. Magn.* **22**(1), 74–86 (2005a)
- Bonchelet, C.G.: Image noise models. In: Bovik, A.C. (eds), *Handbook of Image and Video Processing, Communications, Networking and Multimedia*, Academic Press, Cambridge, pp. 397–410 (2005)
- Zheng, J., Valavanis, K.P., Gauch, J.M.: Noise removal from color images. *J. Intell. Robot. Syst.* **7**(1), 257–285 (1993)
- Faraji, H., MacLean, W.J.: CCD noise removal in digital images. *IEEE Trans. Image Process.* **15**(9), 2676–2685 (2006)
- Liu, C., Szeliski, R., Kang, S., Zitnick, C., Freeman, W.: Automatic estimation and removal of noise from a single image. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(2), 299–314 (2008)
- Huang, Y., Ng, M., Wen, Y.: Fast image restoration methods for impulse and Gaussian noise removal. *IEEE Signal Proc. Lett.* **16**(6), 457–460 (2009)
- Lien, C., Huang, C., Chen, P., Lin, Y.: An efficient denoising architecture for removal of impulse noise in images. *IEEE Trans. Comput.* **62**(4), 631–643 (2013)
- Yang, S.M., Tai, S.C.: A design framework for hybrid approaches of image noise estimation and its application to noise reduction. *J. Vis. Commun. Image Rep.* **23**(5), 812–826 (2012)
- Astola, J., Haavisto, P., Neuvo, Y.: Vector median filters. *Proc. IEEE* **78**(4), 678–689 (1990)
- Celebi, M.E.: Distance measures for reduced ordering-based vector filters. *IET Image Process.* **3**(5), 249–260 (2009). ISSN 1751-9659
- Celebi, M., Kingravi, H., Lukac, R., Celiker, F.: Cost-effective implementation of order-statistics based vector filters using minimax approximations. *J. Opt. Soc. Am. A* **26**(6), 1518–1524 (2009)
- Lukac, R., Smolka, B., Plataniotis, K., Venetsanopoulos, A.: Entropy vector median filter. *Lect. Notes Comput. Sci.* **2652**, 1117–1125 (2003)
- Smolka, B., Malik, K.: Reduced ordering technique of impulsive noise removal in color images. *Lect. Notes Comput. Sci.* **7786**, 296–310 (2013)
- Vertan, C., Malciu, M., Buzuloiu, V., Popescu, V.: Median filtering techniques for vector valued signals. In: *Proceedings of ICIP*, volume I, Lausanne, pp. 977–980 (1996)
- Viero, T., Oistamo, K., Neuvo, Y.: Three-dimensional median-related filters for color image sequence filtering. *IEEE Trans. Circ. 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. Image. Vis.* **23**(3), 315–319 (2005)
- Masoomzadeh-Fard, A., Venetsanopoulos, A.N.: An efficient vector ranking filter for colour image restoration. *Can. Conf. Electr. Comput. Eng.* **2**, 1025–1028 (1993)
- Lukac, R.: Adaptive vector median filtering. *Pattern Recognit. Lett.* **24**(12), 1889–1899 (2003)
- Morillas, S., Gregori, V.: Robustifying vector median filter. *Sensors* **11**(8), 8115 (2011)
- Gregori, V., Morillas, S., Sapena, A.: Adaptive marginal median filter for colour images. *Sensors* **11**(3), 3205–3213 (2011)
- Pitas, I., Venetsanopoulos, A.: *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic Publishers, Boston (1990)

23. Morillas, S., Camacho, J., Latorre, P.: Efficient impulsive noise suppression based on statistical confidence limits. *J. Image Sci. Technol.* **50**(5), 427–436 (2006)
24. Geng, X., Hu, X., Xiao, J.: Quaternion switching filter for impulse noise reduction in color image. *Signal Process.* **92**(1), 150–162 (2012)
25. Jin, L., Xion, C., Liu, H.: Improved bilateral filter for suppressing mixed noise in color images. *Digit. Signal Process.* **22**(6), 903–912 (2012)
26. 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 (2008a)
27. Smolka, B., Plataniotis, K.N., Lukac, R., Venetsanopoulos, A.N.: Similarity based impulsive noise removal in color images. In: *Proceedings of International Conference on Multimedia and Expo, ICME*, vol. 1, pp. 1–585–8 (2003)
28. Aslandogan, Y.A., Celebi, M.E.: Robust switching vector median filter for impulsive noise removal. *J. Electr. Image* **17**, 17–19 (2008)
29. Baljovic, A., Baljovic, D., Kovacevic, B.: Novel method for removal of multichannel impulse noise based on half-space deepest location. *J. Electr. Image* **21**, 21–28 (2012)
30. Baljović, D., Baljović, A., Kovačević, B.: *Impulse and Mixed Multichannel Denoising Using Statistical Halfspace Depth Functions*, pp. 137–194. Springer, Dordrecht (2014)
31. Jin, L., Zhu, Z., Xu, X., Li, X.: Two-stage quaternion switching vector filter for color impulse noise removal. *Signal Process.* **128**, 171–185 (2016). ISSN 0165-1684
32. Aslandogan, Y.A., Celebi, M.E., Kingravi, H.A.: Nonlinear vector filtering for impulsive noise removal from color images. *J. Electr. Image* **16**, 16–21 (2007)
33. Morillas, S., Gregori, V., Sapena, A., Camarena, J., Roig, B.: *Impulsive Noise Filters for Colour Images*, pp. 81–129. Springer, Cham (2015)
34. Ruchay, A., Kober, V.: Impulsive noise removal from color images with morphological filtering. (2017). [arXiv:abs/1707.03126](https://arxiv.org/abs/1707.03126)
35. Peris-Fajarnés, G., Roig, B., Vidal, A.: Rank-Ordered Differences Statistic Based Switching Vector Filter. volume 4141 of *Lecture Notes in Computer Science*, Springer, New York, pp. 74–81 (2006)
36. Burger, W., Burge, M.J.: *Principles of Digital Image Processing: Advanced Methods*. Undergraduate Topics in Computer Science. Springer, New York (2013)
37. 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 Rep.* **17**(1), 1–26 (2006)
38. 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 (2005b)
39. Deng, Y., Kenney, C., Manjunath, B.S.: Peer group image enhancement. *IEEE Trans. Image Process.* **10**(2), 326–334 (2001)
40. Deng, Y., Kenney, C., Moore, M., Manjunath, B.: Peer group filtering and perceptual color image quantization. In: *Proceedings of IEEE International Symposium on Circuits and Systems*, volume 4, pp. 21–24. Springer, New York (1999)
41. Smolka, B., Chydzinski, A.: Fast detection and impulsive noise removal in color images. *Real-Time Image* **11**(5–6), 389–402 (2005)
42. Malinski, L., Smolka, B.: Fast averaging peer group filter for the impulsive noise removal in color images. *J. Real-Time Image Process.* **11**, 427–444 (2015)
43. Jin, L., Liu, H., Xu, X., Song, E.: Quaternion-based color image filtering for impulsive noise suppression. *J. Electr. Image* **19**(4), 043003 (2010)
44. Hu, X., Geng, X.: Quaternion based switching filter for impulse noise removal in color images. *J. Beijing Univ. Aeronaut. Astron.* **9**, 1181 (2012)
45. Wang, G., Liu, Y., Zhao, T.: A quaternion-based switching filter for colour image denoising. *Signal Process.* **102**, 216–225 (2014)
46. Chatzis, V., Pitas, I.: Fuzzy scalar and vector median filters based on fuzzy distances. *IEEE Trans. Image Process.* **8**(5), 731–734 (1999)
47. Yuzhong, S., Barner, K.E.: Fuzzy vector median-based surface smoothing. *IEEE Trans. Vis. Comput. Graph.* **10**(3), 252–265 (2004)
48. Morillas, S., Gregori, V., Peris-Fajarnés, G., Latorre, P.: A new vector median filter based on fuzzy metrics, volume 3656 of *Lecture Notes in Computer Science*, pp. 81–90. Springer, New York (2005)
49. Lukac, R., Plataniotis, K.N., Smolka, B., Venetsanopoulos, A.N.: cDNA microarray image processing using fuzzy vector filtering framework. *J. Fuzzy Sets Syst.* **152**(1), 17–35 (2005c)
50. Camarena, J., Gregori, V., Morillas, S., Sapena, A.: A simple fuzzy method to remove mixed Gaussian-impulsive noise from colour images. *IEEE Trans. Fuzzy Syst.* **21**(5), 971–978 (2013)
51. Morillas, S., Gregori, V., Peris-Fajarnés, G., Sapena, A.: Local self-adaptive fuzzy filter for impulsive noise removal in color images. *Signal Process.* **88**(2), 390–398 (2008b)
52. Plataniotis, K.N., Androustos, D., Venetsanopoulos, A.N.: Color image processing using adaptive vector directional filters. *IEEE Trans. Circ. Syst. II Analog Digit. Signal Process.* **45**(10), 1414–1419 (1998)
53. Gregori, V., Morillas, S., Roig, B., Sapena, A.: Fuzzy averaging filter for impulse noise reduction in colour images with a correction step. *J. Vis. Commun. Image Rep.* **55**, 518–528 (2018). ISSN 1047-3203
54. Malinski, L., Smolka, B.: Fast adaptive switching technique of impulsive noise removal in color images. *J. Real Time Image Process* (2016)
55. Phu, M., Tischer, P., Wu, H.: Statistical analysis of impulse noise model for color image restoration. In: *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)* (2007)
56. Vardavoulia, M.I., Andreadis, I., Tsalides, P.: A new vector median filter for colour image processing. *Pattern Recognit. Lett.* **22**(6), 675–689 (2001)
57. 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)
58. Malinski, L., Smolka, B., Jama, D.: On the efficiency of a fast technique of impulsive noise removal in color digital images. In: *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 855–860 (2017)
59. Malinski, L., Smolka, B.: Training image set (2018a). <https://www.kaggle.com/lmalinski/training-image-set>. Accessed 01 Feb 2019
60. Malinski, L., Smolka, B.: Validation image set (2018b). <https://www.kaggle.com/lmalinski/validation-image-set>. Accessed 01 Feb 2019
61. Lukac, R.: Adaptive color image filtering based on center-weighted vector directional filters. *Multidim. Syst. Signal Process.* **15**(2), 169–196 (2004)
62. Siegel, S., Castellan, N.: *Nonparametric statistics for the behavioral sciences*, 2nd edn. McGraw-Hill Inc, New York (1988)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**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 digital image processing but also he explores field of multidimensional optimisation for nonlinear problems.

**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.