Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Improved switching based filter for protecting thin lines of color images

Chang-cheng WU, Chun-yu ZHAO[†‡], Da-yue CHEN

(*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

[†]E-mail: zhaocy@sjtu.edu.cn

Received Mar. 12, 2009; Revision accepted June 19, 2009; Crosschecked Sept. 29, 2009

Abstract: The classical vector median filter (VMF) has been widely used to remove impulse noise from color images. However, since the VMF cannot identify thin lines during the denoising process, many thin lines may be removed out as noise. This serious problem can be solved by a newly proposed filter that uses a noise detector to find these thin lines and then keep them unchanged. In this new approach, the noise detection scheme applied on a current processed pixel is realized through counting the close pixels in its eight neighbor positions and the expanded window to see whether the current pixel is corrupted by impulse noise. Based on the previous outputs, our algorithm can increase the performance in detecting and canceling the impulse noise. Extensive experiments indicate that this approach can be used to remove the impulse noise from a color image without distorting the useful information.

Key words: Vector median filter (VMF), Thin lines, Impulse noise, Color image
doi:10.1631/jzus.C0910145 **Document code:** A **CLC number:** TP317.4

1 Introduction

Images are often corrupted by impulsive noise due to a noisy sensor or channel transmission errors (Gonzalez and Woods, 2001). For fixed-valued impulse noise, a noisy pixel takes a value of either 0 or 255 (Hwang and Haddad, 1995), while in random-valued impulse noise, noisy intensity is uniformly distributed within the range of [0, 255] (Chen and Wu, 2001). Color image filtering techniques can be divided into the component-wise method and the vector method in terms of their fundamentals. Since the vector method usually considers inherent correlations between color channels, it is widely adopted to protect the chromaticity in denoising color images.

Classical vector filters, including the vector median filter (VMF) (Astola *et al.*, 1990), the basic vector directional filter (BVDF) (Trahanias *et al.*, 1996), and the directional-distance filter (DDF) (Karakos and Trahanias, 1997), are realized through ordering vectors in a sliding filter window. However, these classical filters are implemented uniformly across a whole image and tend to modify both the noise and noise-free pixels. Therefore, the switching based algorithms are proposed to obtain better performance in suppressing the noise and protecting the noise-free pixels, such as the adaptive vector median filter (AVMF) (Lukac, 2003), the selection center weighted vector direction filter (SCWVDF) (Lukac, 2004), the fast peer group filter (FPGF) (Smolka and Chydzinski, 2005), and the switching vector median filter (SVMF) (Jin and Li, 2007). By introducing the switching mechanism to detect whether the pixels are corrupted by impulse noise, the switching based filters restrict the filtering operation on the noisy pixels, thus preventing the removal of fine details and edges. Recently, fuzzy based methods (Lukac *et al.*, 2006; Ma *et al.*, 2006; Celebi *et al.*, 2007; Morillas *et al.*, 2007; Schulte *et al.*, 2007; Camarena *et al.*, 2008) have also been proposed to improve the performance in the whole filtering process. These filters first classify pixels into different signal activity categories and

[‡] Corresponding author

then use a different smoothing method for each category.

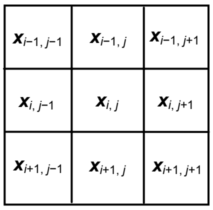
Unfortunately, almost none of the above mentioned filters can distinguish between thin lines and impulse noise. Accordingly, the thin lines are usually interpreted as noise and removed out. In this paper, we consider the detailed information from arbitrary directions in designing our switching mechanism. In addition, since the whole image is processed from left to right and up to down, the previous outputs can be used to improve the performance of detecting the impulse noise and obtaining the noise cancelation results. According to the noise detection results, the noise-free pixels are kept as original values while the noise pixels are canceled by the VMF operations. In the experiments on varieties of testing images, our algorithms can produce significantly better results than other existing filters both in detecting and canceling the impulse noise.

2 Reviewing algorithm of SVMF

Let $\{(i, j)|1\le i\le H,\ 1\le j\le W\}$ be a pixel coordinate of an RGB image, where H and W denote the height and width of the image, respectively. For a filter window, sized $(2N+1)\times(2N+1)$ and centered at (i, j), let $w_{i,j}^{0}=\{\boldsymbol{x}_{i-N,j-N},\cdots,\boldsymbol{x}_{i,j},\cdots,\boldsymbol{x}_{i+N,j+N}\}$ be the samples inside it. Fig. 1 shows the samples in a 3×3 filtering window. Let VMF denote the operation of obtaining the vector median from a set of samples. The output of VMF, $\boldsymbol{y}_{i,j}^{\mathrm{VMF}}$, is defined as

$$\boldsymbol{y}_{i,j}^{\mathrm{VMF}}=\mathrm{VMF}\{w_{i,j}^{0}\}=\underset{\boldsymbol{x}_{s,t}\in w_{i,j}^{0}}{\arg\min}\left\{\sum_{p=-N}^{N}\sum_{q=-N}^{N}\left\|\boldsymbol{x}_{i+p,j+q}-\boldsymbol{x}_{s,t}\right\|_{2}\right\},\tag{1}$$

where $\|\cdot\|_2$ denotes the L_2 norm between two color vectors.



$\boldsymbol{x}_{i-1,j-1}$	$\boldsymbol{x}_{i-1,j}$	$\boldsymbol{x}_{i-1,j+1}$
$\boldsymbol{x}_{i,j-1}$	$\boldsymbol{x}_{i,j}$	$\boldsymbol{x}_{i,j+1}$
$\boldsymbol{x}_{i+1,j-1}$	$\boldsymbol{x}_{i+1,j}$	$\boldsymbol{x}_{i+1,j+1}$

Fig. 1 Samples in the 3×3 filtering window $\boldsymbol{w}_{i,j}^{0}$

The algorithm about the switching based method is often given as

$$\boldsymbol{y}_{i,j}=\boldsymbol{x}_{i,j}\alpha_{i,j}+(1-\alpha_{i,j})\boldsymbol{m}_{i,j},\tag{2}$$

where $\alpha_{i,j}$, valued 1 or 0 for the noise-free pixel and the noisy pixel respectively, is the noise detection result, $\boldsymbol{m}_{i,j}$ is the noise cancelation result, and $\boldsymbol{y}_{i,j}$ is the output value.

The SVMF protects the thin lines and detects the impulse noise in the corrupted color image by comparing the color distance between the current sample and other samples in four directional operators. As shown in Fig. 2, these operators are centered at $\boldsymbol{x}_{i,j}$ and denoted as K_p (p=1, 2, 3, 4). Let the five pixels corresponding to the five nonzero elements in K_p be $\{\boldsymbol{x}_{(p,1)}, \boldsymbol{x}_{(p,2)}, \boldsymbol{x}_{i,j}, \boldsymbol{x}_{(p,3)}, \boldsymbol{x}_{(p,4)}\}$. For the current pixel and the samples in each operator, the average color distance can be obtained as

$$\mathrm{LineDiff}(K_p)=\frac{1}{4}\sum_{l=1}^{4}\mathrm{ColorDiff}(\boldsymbol{x}_{i,j},\boldsymbol{x}_{(p,l)}),\tag{3}$$

where ColorDiff($\boldsymbol{x}_{i,j}$, $\boldsymbol{x}_{(p,l)}$) is the color distance between the two pixel vectors in CIELAB space. The noise detector of the SVMF compares the minimum value of these four directional operators, i.e., min{LineDiff(K_p)} (p=1, 2, 3, 4), with a threshold value. If the minimum average value is less than a set threshold, the current pixel will be detected as noise-free.

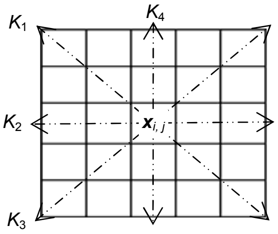
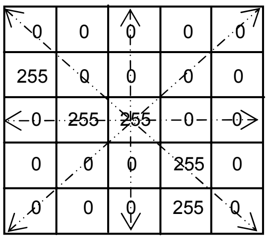


Fig. 2 Four directional operators in SVMF

Seen from the directions of these four operators, they consider only the samples aligned in n×45° (n=0, 1, 2, …, 8). However, the thin lines may be aligned not only in n×45° (n=0, 1, 2, …, 8) directions but also in other arbitrary directions. As shown in Fig. 3, let us consider one case of the thin line, not aligned in n×45° directions. According to the switching mechanism of SVMF, we can see that both K_1 and K_2 can obtain the minimum value amongst these four operators. For the current pixel and the other four samples in each operator K_1 or K_2, since only one sample has a small color distance with the current pixel and the other three samples have a very large color distance, the average color distance will be large enough to make the center pixel recognized as impulsive noise. Therefore, we can say that the SVMF has some drawbacks in protecting the thin lines.



0	0	0	0	0
255	0	0	0	0
0	255	255	0	0
0	0	0	255	0
0	0	0	255	0

Fig. 3 Example about the thin lines with 0=[0, 0, 0] and 255=[255, 255, 255] denoting the color vectors

3 Algorithm about our method

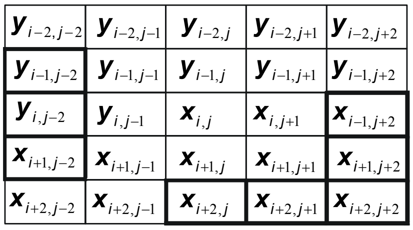
We present a novel noise detection mechanism and an improved noise cancelation method in our algorithm. For the noise detection mechanism, we count the close pixels in eight neighboring positions and the new expanded positions to consider whether the current pixel is corrupted by the impulse noise. For the noise cancelation method, we combine the previous outputs and the later processed pixels as the samples of VMF operation in order to suppress the impulse noise as much as possible.

3.1 Obtaining the expanded window

When the whole image is processed from left to right and up to down, for each current filtering window, half of the positions have been processed and given their outputs while the other half will be processed later. Many switching based algorithms usually use the original samples in the filtering window to detect the impulse noise and give the noise cancelation result. However, if the pixels in the previous processed positions are noise, these noisy pixels will badly affect the detecting and canceling of the impulse noise. Therefore, if these noisy pixels in the previous positions are replaced as their corresponding outputs, we can lessen their bad effects in detecting and canceling the impulse noise. As shown in Fig. 4, for a 5×5 filtering window, let y denote the previous outputs and x denote the later processed samples inside it. For the center pixel $\boldsymbol{x}_{i,j}$, let $\tilde{w}_{i,j}^0$ denote its new eight neighbors and $w_{i,j}^1$ denote the 16 circular samples in the out edge of the 5×5 filtering window as

$$\begin{aligned}\tilde{w}_{i,j}^0 = \{&\boldsymbol{y}_{i-1,j-1}, \boldsymbol{y}_{i-1,j}, \boldsymbol{y}_{i-1,j+1}, \boldsymbol{y}_{i,j-1},\\ &\boldsymbol{x}_{i,j+1}, \boldsymbol{x}_{i+1,j-1}, \boldsymbol{x}_{i+1,j}, \boldsymbol{x}_{i+1,j+1}\},\end{aligned} \tag{4}$$

$$\begin{aligned}w_{i,j}^1 = \{&\boldsymbol{y}_{i-2,j-2}, \boldsymbol{y}_{i-2,j-1}, \boldsymbol{y}_{i-2,j}, \boldsymbol{y}_{i-2,j+1},\\ &\boldsymbol{y}_{i-2,j+2}, \boldsymbol{y}_{i-1,j-2}, \boldsymbol{y}_{i-1j+2}, \boldsymbol{y}_{i,j-2},\\ &\boldsymbol{x}_{i,j+2}, \boldsymbol{x}_{i+1,j-2}, \boldsymbol{x}_{i+1,j+2}, \boldsymbol{x}_{i+2,j-2},\\ &\boldsymbol{x}_{i+2,j-1}, \boldsymbol{x}_{i+2,j}, \boldsymbol{x}_{i+2,j+1}, \boldsymbol{x}_{i+2,j+2}\}.\end{aligned} \tag{5}$$



$\boldsymbol{y}_{i-2,j-2}$	$\boldsymbol{y}_{i-2,j-1}$	$\boldsymbol{y}_{i-2,j}$	$\boldsymbol{y}_{i-2,j+1}$	$\boldsymbol{y}_{i-2,j+2}$
$\boldsymbol{y}_{i-1,j-2}$	$\boldsymbol{y}_{i-1,j-1}$	$\boldsymbol{y}_{i-1,j}$	$\boldsymbol{y}_{i-1,j+1}$	$\boldsymbol{y}_{i-1,j+2}$
$\boldsymbol{y}_{i,j-2}$	$\boldsymbol{y}_{i,j-1}$	$\boldsymbol{x}_{i,j}$	$\boldsymbol{x}_{i,j+1}$	$\boldsymbol{x}_{i-1,j+2}$
$\boldsymbol{x}_{i+1,j-2}$	$\boldsymbol{x}_{i+1,j-1}$	$\boldsymbol{x}_{i+1,j}$	$\boldsymbol{x}_{i+1,j+1}$	$\boldsymbol{x}_{i+1,j+2}$
$\boldsymbol{x}_{i+2,j-2}$	$\boldsymbol{x}_{i+2,j-1}$	$\boldsymbol{x}_{i+2,j}$	$\boldsymbol{x}_{i+2,j+1}$	$\boldsymbol{x}_{i+2,j+2}$

Fig. 4 Samples in $\tilde{\boldsymbol{w}}_{i,j}^0$ and $\tilde{\boldsymbol{w}}_{i,j}^1$

The bold black squares show the expounded window $w_{i,j}^2$ about Fig. 3

For two pixels $\boldsymbol{x}_{s,t}$ and $\boldsymbol{x}_{i,j}$, the close condition between them is that their color distances in RGB space should be less than a threshold d , i.e.,

$$\|\boldsymbol{x}_{i,j} - \boldsymbol{x}_{s,t}\|_2 \le d. \tag{6}$$

Before introducing the expanded window $w_{i,j}^2$, we first need to detect the close pixels in $\tilde{w}_{i,j}^0$ according to Eq. (6) and then obtain $w_{i,j}^2$ through the following steps:

1. Detecting the close pixels of $\boldsymbol{x}_{i,j}$ in its eight neighbors $\tilde{w}_{i,j}^0$ according to Eq. (6).

2. Getting the eight neighbors about one special close pixel according to the structure of Fig. 4.

3. Intersecting the eight neighbors about such special close pixel with $w_{i,j}^1$ to make the intersected samples form the samples of $w_{i,j}^2$.

4. Before all of the close pixels in $\tilde{w}_{i,j}^0$ are processed, repeating steps 2 and 3 and making the intersected samples a union to form the subset of $w_{i,j}^2$.

Let us take the thin line, as shown in Fig. 3, as an example of obtaining the expanded filter window $w_{i,j}^2$. According to Eq. (6), $\boldsymbol{x}_{i+1,j+1}$ and $\boldsymbol{y}_{i,j-1}$ are detected as the close samples to $\boldsymbol{x}_{i,j}$. Seen from the structure of Fig. 4, for pixel $\boldsymbol{x}_{i+1,j+1}$, its eight neighbor samples are $\{\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,j+1}, \boldsymbol{x}_{i,j+2}, \boldsymbol{x}_{i+1,j}, \boldsymbol{x}_{i+1,j+2}, \boldsymbol{x}_{i+2,j}, \boldsymbol{x}_{i+2,j+1}, \boldsymbol{x}_{i+2,j+2}\}$. Thus, the intersected samples with $w_{i,j}^1$ are $\{\boldsymbol{x}_{i,j+2}, \boldsymbol{x}_{i+1,j+2}, \boldsymbol{x}_{i+2,j+2}, \boldsymbol{x}_{i+2,j+1}, \boldsymbol{x}_{i+2,j}\}$. For the other close pixel $\boldsymbol{y}_{i,j-1}$, its eight neighbor samples are $\{\boldsymbol{y}_{i-1,j-2}, \boldsymbol{y}_{i-1,j-1}, \boldsymbol{y}_{i-1,j}, \boldsymbol{y}_{i,j-2}, \boldsymbol{x}_{i,j}, \boldsymbol{x}_{i+1,j-2}, \boldsymbol{x}_{i+1,j-1}, \boldsymbol{x}_{i+1,j}\}$. Thus, the intersected samples with $w_{i,j}^1$ are $\{\boldsymbol{y}_{i-1,j-2}, \boldsymbol{y}_{i,j-2}, \boldsymbol{x}_{i+1,j-2}\}$. The union about all these intersected samples forms the structure of $w_{i,j}^2$.

$$w_{i,j}^2 = \{\boldsymbol{x}_{i,j+2}, \boldsymbol{x}_{i+1,j+2}, \boldsymbol{x}_{i+2,j+2}, \boldsymbol{x}_{i+2,j+1}, \boldsymbol{x}_{i+2,j}, \boldsymbol{x}_{i-1,j-2}, \boldsymbol{x}_{i,j-2}, \boldsymbol{x}_{i+1,j-2}\}. \tag{7}$$

3.2 Noise detection and cancelation

Let n_1 and n_2 denote the number of the close pixels detected in the eight neighbors $\tilde{w}_{i,j}^0$ and in the expounded window $w_{i,j}^2$, respectively. According to Eq. (6), when none of the close pixels can be detected in $\tilde{w}_{i,j}^0$, both n_1 and n_2 should be set to 0. The noise detection result for our algorithm can be described as

$$\alpha_{i,j} = \begin{cases} 1, & \text{Val} \geq \text{Tol}, \\ 0, & \text{Val} < \text{Tol}, \end{cases} \tag{8}$$

where Val is the total number of the close pixels detected in $\tilde{w}_{i,j}^0$ and $w_{i,j}^2$, giving Val=n_1+n_2, and Tol is the threshold number determined by the later experiment. If Val≥Tol, $\boldsymbol{x}_{i,j}$ is considered as noise-free and then kept unchanged; otherwise $\boldsymbol{x}_{i,j}$ is noisy and should be canceled by the VMF operation.

The measurement in Eq. (8) can detect the impulsive noise due to the following reasons:

1. Val=0 when $\boldsymbol{x}_{i,j}$ is an isolated impulsive noise because none of the close pixels can be detected in the eight neighbors $\tilde{w}_{i,j}^0$ with n_1=0 and n_2=0.

2. Val will be large when $\boldsymbol{x}_{i,j}$ is a pixel on a certain line because the thin lines are usually continuous and there will be more close pixels in the eight neighbors $\tilde{w}_{i,j}^0$ and the expounded window $w_{i,j}^2$.

3. Val will also be large when $\boldsymbol{x}_{i,j}$ is neither impulsive noise nor a thin line because more close pixels will be detected in the eight neighbors $\tilde{w}_{i,j}^0$ and the expounded window $w_{i,j}^2$.

As shown in Fig. 3, according to Eq. (6), two close pixels $\boldsymbol{y}_{i-1,j}$ and $\boldsymbol{x}_{i+1,j+1}$ will be detected in the eight neighbors $\tilde{w}_{i,j}^0$ and two close pixels $\boldsymbol{y}_{i-1,j-2}$ and $\boldsymbol{x}_{i+1,j-2}$ will be detected in the expounded window $w_{i,j}^2$ (Eq. (7)). Therefore, as n_1=n_2=2, the center pixel will be protected as noise-free by our algorithm if the threshold Tol is set as 4.

Like the method of detecting the impulse noise, we combine the previous outputs and the later processed pixels as the samples of the VMF operation to obtain the noise cancelation result. In our method, for suppressing the impulse noise as much as possible, we make the samples in $\tilde{w}_{i,j}^0$ applied with the VMF operation to obtain the noise cancelation result $\boldsymbol{m}_{i,j}$ in Eq. (2) as follows:

$$\begin{aligned} m_{i,j} &= \text{VMF}\{\tilde{w}_{i,j}^0\} \\ &= \underset{\boldsymbol{x}_{s,t} \in \tilde{w}_{i,j}^0}{\arg\min} \left\{ \sum_{p=-N}^{N} \sum_{q=-N}^{N} \left\| \boldsymbol{x}_{i+p,j+q} - \boldsymbol{x}_{s,t} \right\|_2 : (p,q) \neq (0,0) \right\}. \end{aligned} \tag{9}$$

From all the above, each step of our algorithm can be described as follows:

1. Using the previous outputs to detect the close pixels in the eight neighbors $\tilde{w}_{i,j}^0$.

2. Obtaining the expounded window $w_{i,j}^2$ based on the close pixels detected in Step 1.

3. Counting the close pixels in $\tilde{w}_{i,j}^0$ and $w_{i,j}^2$ to detect whether $\boldsymbol{x}_{i,j}$ is corrupted by impulse noise.

4. Applying the corrupted pixels with the VMF operations basd on the samples in $\tilde{w}_{i,j}^0$.

4 Simulation and performance analysis

Throughout this work the impulse noise is modeled as Eq. (10) (Plataniotis and Venetsanopoulos, 2000; Zhou and Mao, 2007), where $\boldsymbol{x}^n$ is the noisy pixel, $\{\boldsymbol{x}_\text{R}, \boldsymbol{x}_\text{G}, \boldsymbol{x}_\text{B}\}$ is the noise free color vector, o_t (t=1, 2, 3) are the impulse values, and p and $\sum p$=

($1-p_1-p_2-p_3$ with $\sum_{t=1}^{3} p_t \le 1$) are the degrees of impulse noise contamination.

$$x^n = \begin{cases} \{o_1, x_G, x_B\} & \text{with probability } p_1 p, \\ \{x_R, o_2, x_B\} & \text{with probability } p_2 p, \\ \{x_R, x_G, o_3\} & \text{with probability } p_3 p, \\ \{o_1, o_2, o_3\} & \text{with probability } p \sum p. \end{cases} \tag{10}$$

The quantitative performances of our algorithm are evaluated in three quantitative criteria: the mean absolute error (MAE), the peak signal to noise ratio (PSNR), and the normalized color difference (NCD). These criteria, defined as follows, are related to the image detail preservation, noise suppression, and the color distortions, respectively:

$$\text{MAE} = \frac{1}{3HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \left\| \boldsymbol{y}_{i,j} - \boldsymbol{o}_{i,j} \right\|_1^1, \tag{11}$$

$$\text{PSNR} = 10 \lg \frac{255^2}{\frac{1}{3HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \left\| \boldsymbol{y}_{i,j} - \boldsymbol{o}_{i,j} \right\|_2^2}, \tag{12}$$

$$\text{NCD} = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} \left[(L_{i,j}^y - L_{i,j}^o)^2 + (u_{i,j}^y - u_{i,j}^o)^2 + (v_{i,j}^y - v_{i,j}^o)^2 \right]^{\frac{1}{2}}}{\sum_{i=1}^{H} \sum_{j=1}^{W} \left[(L_{i,j}^o)^2 + (u_{i,j}^o)^2 + (v_{i,j}^o)^2 \right]^{\frac{1}{2}}}, \tag{13}$$

where $\boldsymbol{y}_{i,j}$ and $\boldsymbol{o}_{i,j}$ are the filtered and original desired values at the coordinate of (i, j) respectively, $\|\cdot\|_1$ represents the L_1 norm, $L_{i,j}^y, u_{i,j}^y, v_{i,j}^y$ and $L_{i,j}^o, u_{i,j}^o, v_{i,j}^o$ represent the lightness and the two chrominance values of $\boldsymbol{y}_{i,j}$ and $\boldsymbol{o}_{i,j}$ in the international commission on illumination lightness uniform value (CIELUV) color space, respectively. In general, better filtering performance means a smaller MAE value in protecting the detailed information, a smaller NCD in distorting color information, and a larger PSNR value in suppressing the noise. As shown in Fig. 5, four standard images were experimented throughout the whole paper.

4.1 Parameters choice

Before carrying out the experiments, we simulate the PSNR performances about the Lena and Peppers images, corrupted by impulse noise with different ratios, to obtain the optimal parameter d in Eq. (6) and Tol in Eq. (8). As shown in Figs. 6a, 6b, 7a and 7b, for the fixed- and random-valued impulsive noise by ratios of 5% and 10%, the best PSNR results can be achieved for Tol=3 and $d \in [30, 50]$. For the higher impulse noise by ratios of 15% and 20%, as shown in Figs. 6c, 6d, 7c and 7d, the best PSNR values require Tol=4 and $d \in [40, 60]$ for both the fixed- and random-valued impulse noise corruption. Therefore, in the following experiments, our algorithm adopts Tol=3 and d=40 as the parameters of our noise detection scheme for the noise corruption lower than 10%, otherwise Tol=4 and d=50 for the noise corruption higher than 10%.

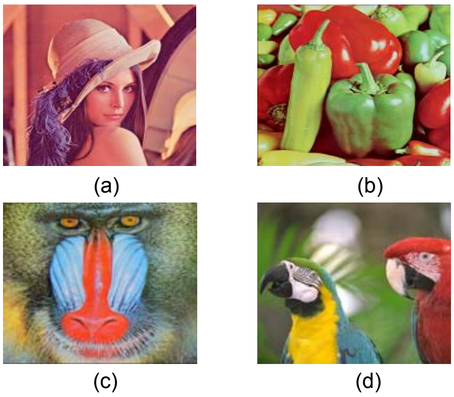


Fig. 5 Testing color images

(a) Lena; (b) Peppers; (c) Baboon; (d) Parrots

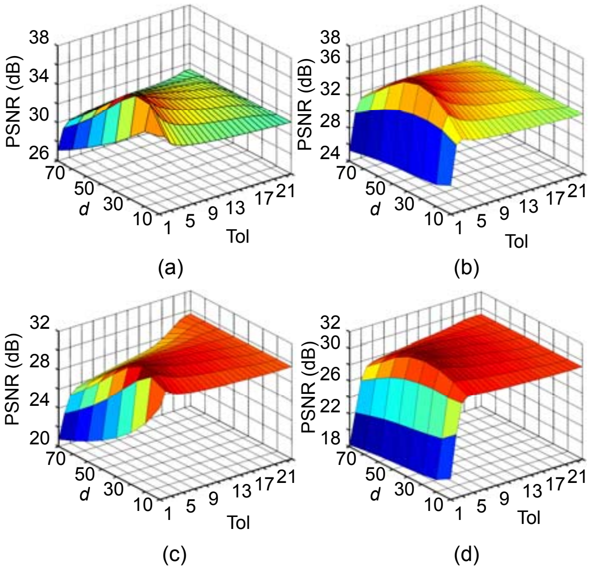


Fig. 6 Parameters *d* and Tol on the PSNR values for image Lena corrupted by the impulse noise

(a) 5% random-valued; (b) 5% fixed-valued; (c) 15% random-valued; (d) 15% fixed-valued

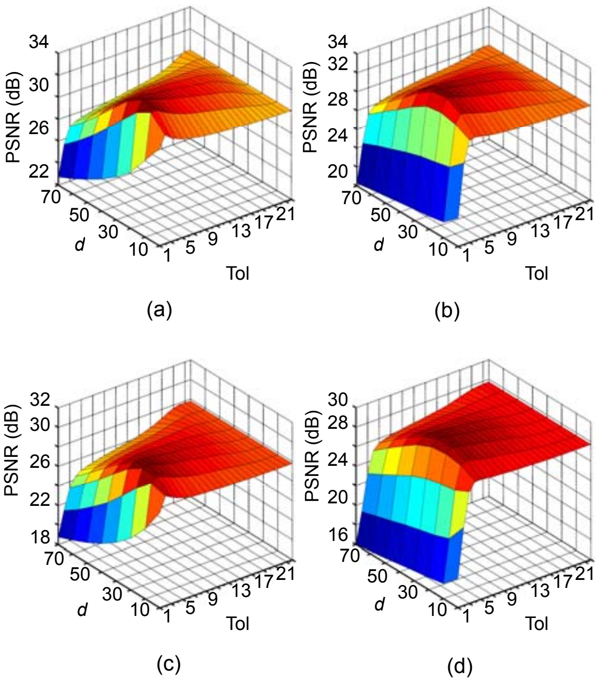


Fig. 7 Parameters *d* and Tol on the PSNR values for image Peppers corrupted by the impulse noise

(a) 10% random-valued; (b) 10% fixed-valued; (c) 20% random-valued; (d) 20% fixed-valued

4.2 Simulation analysis

The quantitative performance of our algorithm is compared with those of several classical filters and switching based filters: the VMF, the BVDF, the DDF (ρ=0.25), the AVMF (r=5, Tol=60), the FPGF (d=45, m=3), the SVMF (Tol=65) and the SCWVDF (λ=2, Tol=0.19). In the whole simulation analysis, the filtering windows about the VMF, BVDF, and DDF are sized 5×5. Tables 1–3 present the PSNR, MAE and NCD values for suppressing the random-valued impulsive noise with the corruption ratio of 5%, 10% and 15%, respectively. As shown in Tables 1 and 2, our algorithm has a much better performance than these classical filters (VMF, BVDF, DDF) in terms of MAE, PSNR and NCD at the low noise corruption ratio. When the noise ratio is high (Table 3), our algorithm still works better than the VMF, BVDF and DDF in the three quantitative aspects. Therefore, we can say that our algorithm could achieve the best quantitative performance among these classical filters for suppressing the impulsive noise and protecting the detailed information. Compared with other switching filters, such as the AVMF, FPGF, SVMF, and SCWVDF, it can also give a comparable performance in PSNR, MAE and NCD values when the noise ratio is low. For a high noise ratio (Table 3), our algorithm can still give much better results in these three quantitative aspects, especially in the PSNR and NCD values. Therefore, our algorithm could achieve the best quantitative performance amongst all of these compared filters.

In order to show the performance of our algorithm in protecting the thin lines while removing the impulse noise, Fig. 8 compares the visual restoration results of our algorithm with several existing filters including the classical filters (VMF, BVDF, DDF) and the switching filters (AVMF, FPGF, SCWVDF, SVMF). As shown in Fig. 8a, the original image Parrots is contaminated by 10% random-valued impulsive noise and attached several thin lines with arbitrary directions. As shown in Figs. 8b–8d, although these classical filters (VMF, BVDF, and DDF) can suppress the noise well, the attached thin lines are almost completely removed. Seen from the images filtered by the AVMF, SCWVDF and FPGF, the thin lines are also suppressed as impulse noise in the whole denoising process. As shown in Fig. 8h, although the square and its two diagonals could be protected by the SVMF, the other two thin lines of arbitrary directions are removed out as noise. These results of SVMF further show its inability in protecting the thin lines in arbitrary directions except for the n×45° (n=0, 1, …, 8). Therefore, we can see that our algorithm works best amongst these compared classical and switching filters in protecting the thin lines of arbitrary directions.

The following experiments examine the performance of our algorithm on real images. Fig. 9 shows two digitized artwork images obtained from the Kodak company. Fig. 10 shows the enlarged parts of the real images and the filtered images of our algorithm and the compared methods. Detailed visual inspection of the restoration results depicted in Fig. 10 reveals that our algorithm can provide an excellent trade-off between the noise attenuation and detail preservation (Figs. 10i and 10j. Unlike our algorithm, owning to the excessive smoothing, the well known classic algorithms (VMF, BVDF, and DDF) blur edges and remove fine details (Figs. 10c–10h).

Table 1 Quantitative restoration results of our algorithm and other methods for the images corrupted by 5% random-valued impulse noise

Filter	Parrots			Lena			Peppers			Baboon		
	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD
VMF	27.11	4.11	0.0511	30.19	4.57	0.0593	30.51	4.74	0.0578	20.74	15.66	0.1725
BVDF	25.36	4.67	0.0602	28.88	5.25	0.0637	27.53	5.96	0.0697	18.07	20.55	0.1894
DDF	27.21	4.21	0.0499	30.16	4.61	0.0594	30.43	4.76	0.0577	20.68	15.74	0.1712
AVMF	29.87	1.58	0.0231	33.27	1.03	0.0216	33.19	1.06	0.0182	25.15	4.32	0.0512
SCWVDF	30.11	1.42	0.0197	33.76	1.27	0.0171	30.62	1.78	0.0259	24.91	6.31	0.0861
FPGF	30.21	1.21	0.0215	34.72	1.02	0.0173	34.59	1.04	0.0154	23.52	7.46	0.0824
SVMF	29.76	1.65	0.0247	32.15	1.14	0.0248	32.07	1.19	0.0209	25.27	3.59	0.0479
Ours	31.62	1.18	0.0119	36.18	0.95	0.0141	35.95	0.97	0.0142	26.31	2.84	0.0355

AVMF: adaptive vector median filter; BVDF: basic vector directional filter; DDF: directional-distance filter; FPGF: fast peer group filter; MAE: mean absolute error; NCD: normalized color difference; PSNR: peak signal to noise ratio; SCWVDF: selection center weighted vector direction filter, SVMF: switching vector median filter; VMF: vector median filter

Table 2 Quantitative restoration results of our algorithm and other methods for the images corrupted by 10% random-valued impulse noise

Filter	Parrots			Lena			Peppers			Baboon		
	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD
VMF	26.71	4.33	0.0612	29.86	4.82	0.0621	29.95	5.03	0.0611	20.65	15.94	0.1766
BVDF	25.16	4.77	0.0715	28.77	5.47	0.0647	26.31	6.41	0.0759	18.01	20.81	0.1925
DDF	26.69	4.38	0.0601	29.86	4.83	0.0618	29.97	5.02	0.0607	20.62	15.98	0.1747
AVMF	27.87	2.44	0.0311	30.43	1.98	0.0413	30.23	2.08	0.0354	24.04	5.91	0.0796
SCWVDF	27.91	2.52	0.0325	30.23	2.33	0.0331	30.06	3.12	0.0467	23.09	8.51	0.1107
FPGF	28.21	2.21	0.0275	32.29	1.87	0.0317	31.95	1.97	0.0291	22.85	9.09	0.1066
SVMF	27.76	2.65	0.0347	30.36	2.22	0.0469	30.31	2.28	0.0393	24.41	5.75	0.0807
Ours	29.89	1.71	0.0261	34.69	1.65	0.0232	32.47	1.74	0.0244	25.07	3.64	0.0487

AVMF: adaptive vector median filter; BVDF: basic vector directional filter; DDF: directional-distance filter; FPGF: fast peer group filter; MAE: mean absolute error; NCD: normalized color difference; PSNR: peak signal to noise ratio; SCWVDF: selection center weighted vector direction filter, SVMF: switching vector median filter; VMF: vector median filter

Table 3 Quantitative restoration results of our algorithm and other methods for the images corrupted by 15% random-valued impulse noise

Filter	Parrots			Lena			Peppers			Baboon		
	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD	PSNR (dB)	MAE	NCD
VMF	26.22	4.61	0.0667	29.49	5.11	0.0653	29.33	5.39	0.0649	20.56	16.25	0.1813
BVDF	24.56	5.17	0.0775	28.11	5.83	0.0681	25.52	6.92	0.0821	17.96	20.99	0.1967
DDF	26.16	4.72	0.0701	29.43	5.12	0.0651	29.32	5.38	0.0649	20.52	16.32	0.1796
AVMF	25.67	3.84	0.0541	29.46	2.99	0.0615	29.22	3.15	0.0533	23.11	7.44	0.1074
SCWVDF	25.72	3.52	0.0495	29.95	3.69	0.0561	29.34	4.75	0.0734	21.62	10.73	0.1311
FPGF	26.01	3.21	0.0435	30.36	2.78	0.0472	29.82	3.02	0.0442	22.23	10.64	0.1302
SVMF	25.76	3.61	0.0515	29.72	3.25	0.0671	29.58	3.39	0.0575	23.13	7.84	0.1118
Ours	27.77	3.31	0.0444	32.01	2.11	0.0389	31.71	2.02	0.0311	24.22	4.41	0.0611

AVMF: adaptive vector median filter; BVDF: basic vector directional filter; DDF: directional-distance filter; FPGF: fast peer group filter; MAE: mean absolute error; NCD: normalized color difference; PSNR: peak signal to noise ratio; SCWVDF: selection center weighted vector direction filter, SVMF: switching vector median filter; VMF: vector median filter

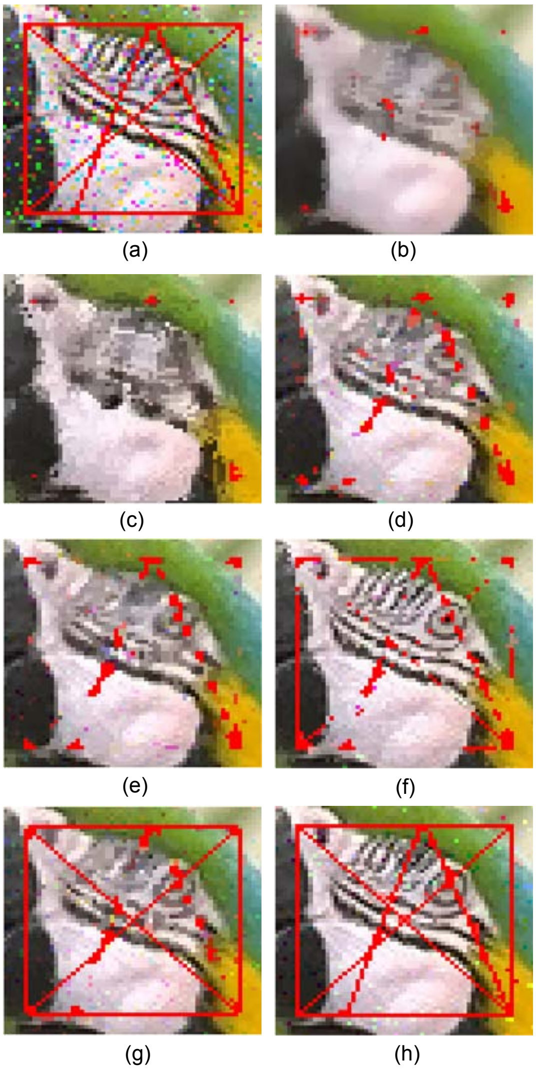


Fig. 8 Visual restoration results of our algorithm and other methods on the image Parrots
(a) Image contaminated by 10% random-valued impulsive noise and attached with several crossed thin lines; (b)–(h) Images filtered by VMF, BVDF, AVMF, FPGF, SWVDF, SVMF, and our algorithm, respectively

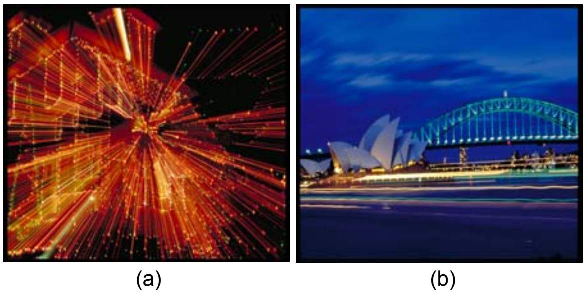


Fig. 9 Real images obtained from the Kodak company
(a) Xmablur; (b) Sydney

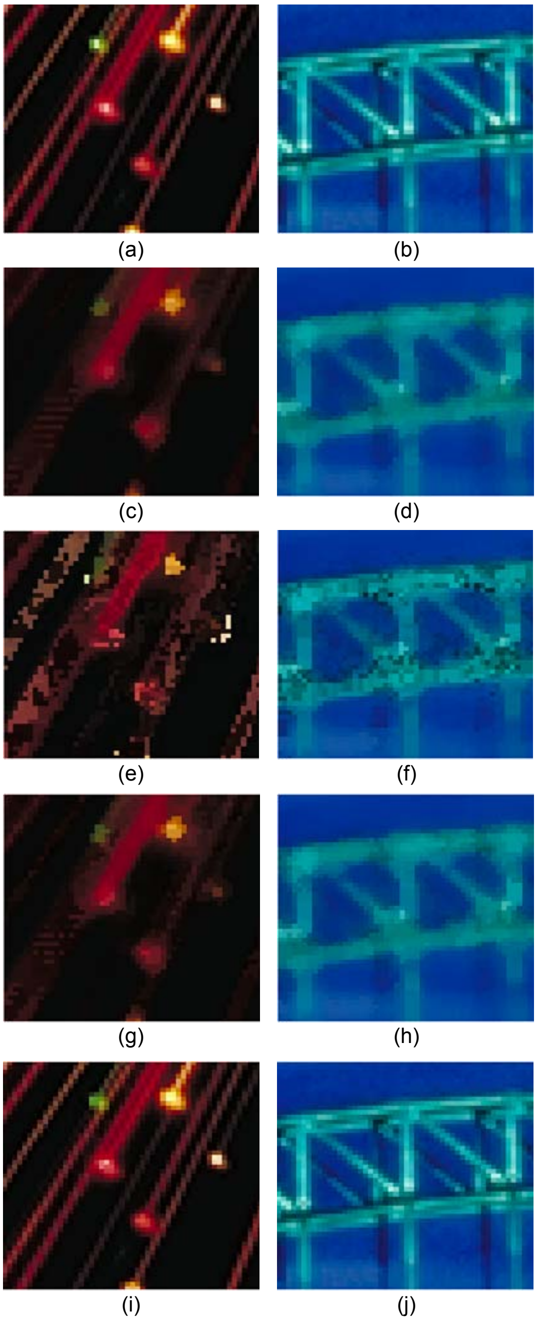


Fig. 10 Visual restoration results of our algorithm and other methods on the real images
(a) and (b) are original real images; (c) and (d), (e) and (f), (g) and (h), and (i) and (j) are images filtered by VMF, BVDF, DDF, and our algorithm, respectively

5 Conclusion

A new switching filter is proposed in this work to suppress the impulsive noise and protect the thin lines at the same time. Based on the previous output, our algorithm detects the close samples in the eight neighbor positions and in the expanded positions to determine whether the central pixel is contaminated by noise. Compared with the SVMF method, our algorithm works better in protecting the thin lines of arbitrary directions. Compared with the classical and other switching filters, since our noise detector and noise cancelation method are based on previous outputs, our algorithm has an excellent performance in both the quantitative and the qualitative restoration aspects.

References

Astola, J., Haavisto, P., Neuvo, Y., 1990. Vector median filter. *Proc. IEEE*, **78**(4):678-689. [doi:10.1109/5.54807]

Camarena, J.G., Gregori, V., Morillas, S., Sapena, A., 2008. Fast detection and removal of impulsive noise using peer groups and fuzzy metrics. *J. Vis. Comm. Image Represent.*, **19**(1):20-29. [doi:10.1016/j.jvcir.2007.04.003]

Celebi, M.E., Kingravi, H.A., Aslandogan, Y.A., 2007. Nonlinear vector filtering for impulsive noise removal from color images. *J. Electron. Imag.*, **16**(3):033008. [doi:10.1117/1.2772639]

Chen, T., Wu, H.R., 2001. Space variant median filters for the restoration of impulse noise corrupted images. *IEEE Trans. Circ. Syst. II: Anal. Dig. Signal Process.*, **48**(8): 784-789. [doi:10.1109/82.959870]

Gonzalez, R.C., Woods, R.E., 2001. Digital Image Processing (2nd Ed.). Prentice-Hall, Englewood Cliffs, NJ.

Hwang, H., Haddad, R.A., 1995. Adaptive median filters: new algorithms and results. *IEEE Trans. Signal Process.*, **4**(4): 499-502. [doi:10.1109/83.370679]

Jin, L.H., Li, D.H., 2007. A switching vector median filter based on the CIELAB color space for color image restoration. *Signal Process.*, **87**(6):1345-1354. [doi:10.1016/j.sigpro.2006.11.008]

Karakos, D.G., Trahanias, P.E., 1997. Generalized multichannel image-filtering structures. *IEEE Trans. Image Process.*, **6**(7):1038-1045. [doi:10.1109/83.597278]

Lukac, R., 2003. Adaptive vector median filtering. *Pattern Recogn. Lett.*, **24**(12):1889-1899. [doi:10.1016/S0167-8655(03)00016-3]

Lukac, R., 2004. Adaptive color image filtering based on center-weighted vector directional filters. *Multidim. Syst. Signal Process.*, **15**(2):169-196. [doi:10.1023/B:MULT.0000017024.66297.a0]

Lukac, R., Smolka, B., Plataniotis, K.N., Venetsanopoulos, A.N., 2006. Vector sigma filters for noise detection and removal in color images. *J. Vis. Commun. Image Represent.*, **17**(1):1-26. [doi:10.1016/ j.jvcir.2005.08.007]

Ma, Z.H., Wu, H.R., Feng, D.G., 2006. Partition-based vector filtering technique for suppression for noise in digital color images. *IEEE Trans. Image Process.*, **15**(8): 2324-2342. [doi:10.1109/TIP.2006.877066]

Morillas, S., Gregori, V., Peris-Fajarnes, G., Sapena, A., 2007. New adaptive vector filter using fuzzy metrics. *J. Electron. Imag.*, **16**(3):033007. [doi:10.1117/1.2767335]

Schulte, S., de Witte, V., Nachtegael, M., van der Weken, D., Kerre, E.E., 2007. Histogram-based fuzzy color filter for image restoration. *Image Vis. Comput.*, **25**(9):1377-1390. [doi:10.1016/j.imavis.2006.10.002]

Smolka, B., Chydzinski, A., 2005. Fast detection and impulse noise removal in color image. *Real-Time Imag.*, **11**(5-6):389-402. [doi:10.1016/j.rti.2005.07.003]

Trahanias, P.E., Karakos, D., Venetsanopoulos, A.N., 1996. Directional processing of color image: theory and experimental results. *IEEE Trans. Image Process.*, **5**(6): 868-880. [doi:10.1109/83.503905]

Plataniotis, K.N., Venetsanopoulos, A.N., 2000. Color Image Processing and Application. Springer, Berlin, p.51-58.

Zhou, H., Mao, K.Z., 2007. An impulse noise color image filter using learning based color morphological operation. *Dig. Signal Process.*, **18**(3):406-421. [doi:10.1016/j.dsp.2007.04.013]