

Two Effective Algorithms for Color Image Denoising

Jian-jun Zhang^(✉), Jian-li Zhang, and Meng Gao

Department of Mathematics, Shanghai University, Shanghai 200444, China
jjzhang@staff.shu.edu.cn

Abstract. We present two effective algorithms for removing impulse noise from color images. Our proposed algorithms take a two-step approach: in the first step, noise color pixel candidates are identified by an impulse detector, and in the second step, only those identified noise candidates in the image are restored by using a modified weighted vector median filter. Extensive experiments indicate that our proposed algorithms have good performance, and are more effective than most of the existing algorithms in removing impulse noise from color images.

Keywords: Color image · Denoising · Impulse noise · Vector median filter

1 Introduction

Removing impulse noise is an import and challenging problem in image processing. A lot of researches have been done for this problem. For example, nonlinear filters such as median filter are successfully applied to impulse noise removal in scalar valued images [6]. For a color image, its pixel has three scalar values. Suppose it is encoded by red, green and blue values in RGB color space. Then it is natural to apply the traditional nonlinear filtering techniques to each channel separately [2]. However, such technique does not use the correlation exists among the three different color channels, and therefore may lead to problems that are not present in scalar valued images [1]. Therefore vector processing techniques for color image denoising are desirable.

To our knowledge, vector median filter (VMF) [1] may be the first and the best-known vector filtering technique for its simplicity, robustness and impulse noise removing ability [13]. Since the development of VMF, many vector filtering techniques are proposed. For example, the vector directional filters (VDF) which consider the vectors' direction is proposed [17]. In [10], the directional-distance filters (DDF) is developed. In [7], a hybrid directional filter (HDF) is proposed by using the output of the VDF and the VMF. Similar to the scalar median filter, the above mentioned vector filters have some undesirable side effects that tend to smear sharp edges and fine details of an image [8]. In order to eliminate such defects, some new vector processing techniques have been developed, see for

example [11, 12] and references therein. We remark here that, relaxation Labeling [4] may also be applied to solving color image denoising problem.

Recently, a two-step approach for removing impulse noise from scalar valued images was developed [3, 5]. In this work, we present two effective algorithms for removing impulse noise from color images. Our proposed algorithms take a two-step approach. In the first step, noise color pixel candidates are identified by an impulse detector, and then in the second step, only those identified noise candidates in the image are restored by using a modified weighted vector median filter. Extensive experiments indicate that our proposed algorithms have good performance, and are more effective than most of the existing algorithms in removing impulsive noise from color images.

The remainder of the paper is organized as follows. In Sect. 2, we clarify the problems we are considered. In Sect. 3, we present our denoising algorithms. Experimental results are presented in Sect. 4 and Sect. 5 concludes the paper.

2 Problem Setting

The problem we are considered is the impulse noise removal in color images. Given an M by N noisy image, let x_{ij} be the vector of the given image, let v_{ij} be the impulse noise vector, and let z_{ij} be the noise-free color pixel, for $(i, j) \in \mathcal{I} \equiv \{1, \dots, M\} \times \{1, \dots, N\}$, and let p be impulse noise probability. Then impulse noise model is described as,

$$x_{ij} = \begin{cases} v_{ij} & \text{with probability } p \\ z_{ij} & \text{with probability } 1 - p \end{cases} \quad (1)$$

The model (1) gives many kinds of impulse noise for color images. Depending on the type of vector v_{ij} , we consider two impulse noise models, called salt-and-pepper noise model, and random-valued impulse noise model. See [9] for example.

For salt-and-pepper noise model, v_{ij} is characterized by the following expression,

$$v_{ij} = \begin{cases} (d_1, v_{ij}^G, v_{ij}^B) & \text{with probability } p_1 \\ (v_{ij}^R, d_2, v_{ij}^B) & \text{with probability } p_2 \\ (v_{ij}^R, v_{ij}^G, d_3) & \text{with probability } p_3 \\ (d_4, d_5, d_6) & \text{with probability } p_4 \end{cases} \quad (2)$$

where d_k takes an extreme value for $k = 1, 2, \dots, 6$, and $p_1 + p_2 + p_3 + p_4 = 1$. In the remainder of the paper, the two extreme values are denoted by x_{\min} and x_{\max} , respectively.

In the case of random-valued impulse noise model, we define v_{ij} by using the following model:

$$v_{ij} = \begin{cases} (r_1, v_{ij}^G, v_{ij}^B) & \text{with probability } p_1 \\ (v_{ij}^R, r_2, v_{ij}^B) & \text{with probability } p_2 \\ (v_{ij}^R, v_{ij}^G, r_3) & \text{with probability } p_3 \\ (r_4, r_5, r_6) & \text{with probability } p_4 \end{cases} \quad (3)$$

where r_k is independent uniformly distributed random numbers for $k = 1, 2, \dots, 6$, and $p_1 + p_2 + p_3 + p_4 = 1$.

In the following, we will design two denoising algorithms based on the presented noise models.

3 Our Proposed Method

As we have known, it should consider the correlations that exists among the different color channels in the processing of color image data [1, 8, 14]. However, this correlation property does not need to be considered when detecting impulses in color images. Since for a color vector $X_i = (X_i^R, X_i^G, X_i^B)$, if one of its components is corrupted by impulse, then the whole vector would be contaminated by the impulse. Therefore we can detect impulses in color images by using impulse detection algorithms for scalar valued images.

In the following, we first present the adaptive median filter (AMF) [6], which will be used to identify salt-and-pepper noise in scalar valued images. Then we present our denoising algorithm.

Let W_{ij}^s be a window of size $(2s + 1) \times (2s + 1)$ centered at location (i, j) . More definitely, $W_{ij}^s = \{(k, l) \mid (k, l) \in \mathcal{I}, |k - i| \leq s \text{ and } |l - j| \leq s\}$. The observed color image vectors in this window consisting of samples denoted by $X = \{X_1, X_2, \dots, X_S\}$, with $x_{ij} = X_{(S+1)/2}$, and $S = (2s+1)^2$. The corresponding three individual color channel are denoted as $X^R = \{X_1^R, X_2^R, \dots, X_S^R\}$, $X^G = \{X_1^G, X_2^G, \dots, X_S^G\}$ and $X^B = \{X_1^B, X_2^B, \dots, X_S^B\}$, respectively. We use X^C to represent either of the X^R, X^G, X^B .

Algorithm 1 (AMF for scalar valued images)

Initialize $s = 1$, and set the maximum window size be s_{\max} . For all pixel location (i, j) , do

1. Compute $x_{\max}^C, x_{\text{med}}^C, x_{\min}^C$, which are respectively the maximum, median and minimum of the pixel values in W_{ij}^s .
2. If $x_{\text{med}}^C = x_{\max}^C$ or $x_{\text{med}}^C = x_{\min}^C$, set $s = s + 1$; otherwise, go to step 4.
3. If $s > s_{\max}$, replace x_{ij}^C by x_{med}^C ; otherwise, go to step 1.
4. If $x_{\min}^C < x_{ij}^C < x_{\max}^C$, then x_{ij}^C is noise-free; replace x_{ij}^C by x_{med}^C , otherwise.

Based on the color image noise model (1)(2), we can identify salt-and-pepper noise in color images by using Algorithm 1.

Algorithm 2 (Noise identification)

Let y^C be the image obtained by applying Algorithm 1 to the channel x^C . Since noisy pixels take only extreme values x_{\min} or x_{\max} , the noise candidate set can be obtained by

$$\mathcal{N} = \{(i, j) \mid (i, j) \in \mathcal{I}, y_{ij}^C \neq x_{ij}^C \text{ and } x_{ij}^C = x_{\min}, \text{ or } x_{ij}^C = x_{\max}\}.$$

We use

$$\mathcal{N}^c = \{(i, j) \mid (i, j) \in \mathcal{I}, \text{ and } (i, j) \notin \mathcal{N}\},$$

to denote the set of noise-free pixels.

By using Algorithm 2, we now present our denoising algorithm for removing salt-and-pepper noise from color images.

Algorithm 3 (*Denoising algorithm for color images corrupted by salt-and-pepper noise*)

1. (*Noise identification*): Dividing the image vectors into noise-free set \mathcal{N}^c and noise corrupted set \mathcal{N} by using Algorithm 2.
2. (*Restoration*): Let the observed image vectors in S_{ij}^w consisting of samples denoted by $X = \{X_1, X_2, \dots, X_S\}$, with $x_{ij} = X_{(S+1)/2}$. If $x_{ij} \in \mathcal{N}^c$, the output is x_{ij} itself. Otherwise, the output is defined as,

$$X_{(1)} = \arg \min_{X_k \in X} \sum_{\substack{l=1 \\ l \in \mathcal{N}^c}}^W \|X_k - X_l\|_2,$$

where $l \in \mathcal{N}^c$ denotes that $(i, j) \in \mathcal{N}^c$ for some $x_{i,j} = X_l$,

$$\|X_k - X_l\|_2 = \sqrt{(X_k^R - X_l^R)^2 + (X_k^G - X_l^G)^2 + (X_k^B - X_l^B)^2},$$

and $\arg \min$ is argument of the minimum, that is,

$$\sum_{\substack{l=1 \\ l \in \mathcal{N}^c}}^S \|X_{(1)} - X_l\|_2 = \min_{X_k \in X} \sum_{\substack{l=1 \\ l \in \mathcal{N}^c}}^S \|X_k - X_l\|_2.$$

We can easily see that our proposed algorithm (Algorithm 3) is easy to implement. In the used distance measure, we give zeros weights on the noise color pixels. This strategy ensures more effective impulse removal than other methods in the literature such as VMF.

We now consider to remove random-valued impulse noise from color images. Here we use the rank-ordered logarithmic difference (ROLD) algorithm developed in [5] to identify noise color pixels.

Algorithm 4 (*ROLD algorithm*)

Let the observed color image vectors in W_{ij}^s consisting of samples denoted by $X = \{X_1, X_2, \dots, X_S\}$, with $x_{ij} = X_{(S+1)/2}$. Let $D_j(X_{(S+1)/2}^C) = 1 + \max\{\log_2 |X_j^C - X_{(S+1)/2}^C|, -5\}/5$, for $j = 1, 2, \dots, S, j \neq (S + 1)/2$. Arrange all D_j in an increasing order,

$$D_{(1)} \leq D_{(2)} \leq \dots \leq D_{(S-1)}.$$

Define local image order statistics as

$$ROLD_l(X_{(S+1)/2}^C) = \sum_{k=1}^l D_{(k)}(X_{(S+1)/2}^C).$$

We then identify random-valued impulse noise candidates by using ROLD with threshold T^C : a color vector $X_{(S+1)/2}$ or equivalently, x_{ij} is corrupted with impulse noise if one of its component $X_{(S+1)/2}^C$ satisfies $ROLD_l(X_{(S+1)/2}^C) > T^C$, and it is noise free otherwise.

Now we present our denoising algorithm for color images contaminated by random-valued impulse noise.

Algorithm 5 (*Denoising algorithm for color images contaminated by random valued impulse noise.*)

1. (*Noise identification*): Dividing the image vectors into noise free set \mathcal{N}^c and noise corrupted set \mathcal{N} by using Algorithm 4.

2. (*Restoration*): Let the observed image vectors in W_{ij}^s consisting of samples denoted by $X = \{X_1, X_2, \dots, X_S\}$, with $x_{ij} = X_{(S+1)/2}$. If $x_{ij} \in \mathcal{N}^c$, the output is x_{ij} itself. Otherwise, the output is given by,

$$X_{(1)} = \arg \min_{X_k \in X} \sum_{\substack{l=1 \\ l \in \mathcal{N}^c}}^S \|X_k - X_l\|_2.$$

4 Experimental Results

We conduct two experiments in this section, to exam the noise removal ability of our proposed algorithms. Three 512×512 color images are used in our experiments. These three images are ‘‘Lena’’, ‘‘Mandrill’’, and ‘‘Lake’’, which are shown in Fig. 1. We also compared performance of our algorithms with many other algorithms in the literature, including VMF [1], AVMF [12], MAVMF [12], quaternion based algorithm, denoted by QVMF [8], VDF [17], AVDF [15], DDF [10] and HDF [7].

In the experiments, 3×3 filter window is used for all above-mentioned techniques. For AVMF, we set $\lambda_1 = 4$, and for MAVMF, we set $\lambda_2 = 12$ [12]. For QVMF, we set the parameter Tol to be 22 as it was set in [8]. For DDF, we set $p = 0.75$ [10]. For AVDF, we use AVDF2, since it performances best in all AVDFs proposed in [15].

We use AMF of maximum size 13 to identify salt-and-pepper noise for Algorithm 3.

For Algorithm 5, we use ROLD algorithm to identify noise as described. We set $l = 4$ and set the window size to be 3×3 , and the threshold is set to be $T = s \cdot q$ with $s = 1.9$ if noise ratio is equal or less than 25%; we set $l = 12$, and set the window size to be 5×5 , and the threshold is set to be $T = s \cdot q$ with $s = 5.4$ if noise ratio is greater than 25%. Where q is the fraction of the pixels in each color channel whose ROLD values are less than s .

To get quantitative measure for the noise removal ability of above mentioned algorithms, the normalized color difference (NCD) and the normalized mean square error (NMSE) are used in this section. Let M, N be the image dimensions, and x_{ij} and \tilde{x}_{ij} be the original image vector and its filtered image vector at location (i, j) , respectively. Let L^* and (u^*, v^*) be lightness values and chrominance values corresponding to x_{ij} and \tilde{x}_{ij} samples encoded in CIE $L^*u^*v^*$ color

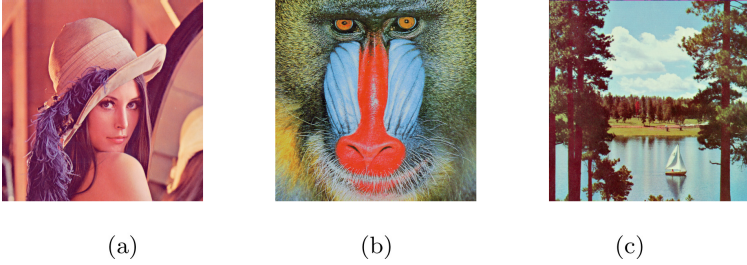


Fig. 1. Original images. (a) Lena, (b) Mandril, (c) Lake.

space, respectively. Then NCD is defined as [16],

$$\text{NCD} = \frac{\sum_{i=1}^M \sum_{j=1}^N \sqrt{(L_{x_{ij}}^* - L_{\tilde{x}_{ij}}^*)^2 + (u_{x_{ij}}^* - u_{\tilde{x}_{ij}}^*)^2 + (v_{x_{ij}}^* - v_{\tilde{x}_{ij}}^*)^2}}{\sum_{i=1}^M \sum_{j=1}^N \sqrt{(L_{x_{ij}}^*)^2 + (u_{x_{ij}}^*)^2 + (v_{x_{ij}}^*)^2}},$$

and NMSE is defined as

$$\text{NMSE} = \frac{\sum_{i=1}^M \sum_{j=1}^N \|x_{ij} - \tilde{x}_{ij}\|_2^2}{\sum_{i=1}^M \sum_{j=1}^N \|x_{ij}\|_2^2}.$$

Experiment 1. In this experiment, we test the salt-and-pepper noise removal ability of the above mentioned algorithms. To this end, we use the noise model (1)(2) to corrupt the three original test images by salt-and-pepper noise. In the corruption, different noise levels ranging from 5% to 35% with increments of 10% are carried out. Tables 1, 2 and 3 show respectively, the values of NCD and NMSE for the restored images “Lena”, “Mandrill” and “Lake”. Where “Alg3” denotes our proposed algorithm (Algorithm 3). In Fig. 2, we show the restored images by applying VMF, AVMF and Alg5 to the 25% noise corrupted Lena image.

From the Tables 1, 2 and 3 and Fig. 2, we see that Algorithm 3, has lowest NCD and NMSE values and good subjective performance. Moreover, our algorithm is easy to implement. Based on the given results, we draw a conclusion that our proposed algorithm, Algorithm 3, is competitive with most existing algorithms when applied to salt-and-pepper noise removal.

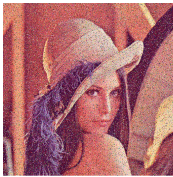
Experiment 2. In this experiment, we test the random-valued impulse noise removal ability of the above mentioned algorithms. To this end, we use the noise

Table 1. Values of NCD and NMSE obtained by different algorithms applied to salt-and-pepper noise corrupted Lena image.

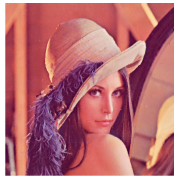
Noise level	5%		15%		25%		35%	
Algorithm	NCD	NMSE	NCD	NMSE	NCD	NMSE	NCD	NMSE
VMF	0.0301	0.0016	0.0326	0.0019	0.0359	0.0024	0.0401	0.0036
AVMF	0.0065	0.0005	0.0101	0.0011	0.0210	0.0046	0.0424	0.0136
MAVMF	0.0272	0.0015	0.0310	0.0018	0.0351	0.0024	0.0398	0.0035
QVMF	0.0029	0.0003	0.0091	0.0012	0.0176	0.0026	0.0281	0.0048
VDF	0.0410	0.0091	0.0705	0.0299	0.1068	0.0565	0.1525	0.0901
AVDF	0.0851	0.0550	0.1862	0.1579	0.2781	0.2487	0.3653	0.3335
DDF	0.0428	0.0093	0.0687	0.0296	0.1046	0.0558	0.1492	0.0882
HDF	0.0393	0.0083	0.0633	0.0248	0.0942	0.0445	0.1318	0.0676
Alg3	0.0021	0.0001	0.0065	0.0004	0.0114	0.0008	0.0172	0.0017

Table 2. Values of NCD and NMSE obtained by different algorithms applied to salt-and-pepper noise corrupted Lena image.

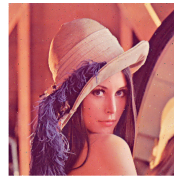
Noise level	5%		15%		25%		35%	
Algorithm	NCD	NMSE	NCD	NMSE	NCD	NMSE	NCD	NMSE
VMF	0.1185	0.0228	0.1268	0.0246	0.1322	0.0269	0.1388	0.0302
AVMF	0.0484	0.0085	0.0557	0.0110	0.0738	0.0189	0.1035	0.0341
MAVMF	0.1163	0.0228	0.1267	0.0246	0.1321	0.0269	0.1388	0.0302
QVMF	0.0603	0.0150	0.0866	0.0304	0.1116	0.0435	0.1349	0.0551
VDF	0.1650	0.0551	0.1954	0.0845	0.2339	0.1210	0.2787	0.1643
AVDF	0.1771	0.0767	0.2681	0.1765	0.3545	0.2713	0.4348	0.3565
DDF	0.1418	0.0380	0.1701	0.0616	0.2075	0.0936	0.2499	0.1305
HDF	0.1309	0.0291	0.1544	0.0468	0.1843	0.0689	0.2178	0.0930
Alg3	0.0076	0.0016	0.0399	0.0049	0.0544	0.0088	0.0702	0.0135



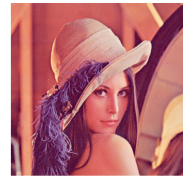
(a)



(b)



(c)



(d)

Fig. 2. Restored images by using different algorithms. (a) 25% salt-and-pepper noise corrupted Lena image, (b) VMF, (c) AVMF, (d) Alg3.

Table 3. Values of NCD and NMSE obtained by different algorithms applied to salt-and-pepper noise corrupted Lena image.

Noise level	5%		15%		25%		35%	
Algorithm	NCD	NMSE	NCD	NMSE	NCD	NMSE	NCD	NMSE
VMF	0.0611	0.0051	0.0714	0.0058	0.0753	0.0069	0.0799	0.0085
AVMF	0.0334	0.0015	0.0387	0.0030	0.0521	0.0076	0.0749	0.0175
MAVMF	0.0602	0.0051	0.0711	0.0058	0.0751	0.0069	0.0799	0.0085
QVMF	0.0380	0.0054	0.0522	0.0136	0.0677	0.0222	0.0844	0.0315
VDF	0.0865	0.0165	0.1146	0.0396	0.1518	0.0705	0.1977	0.1091
AVDF	0.1155	0.0589	0.2125	0.1581	0.3063	0.2562	0.3883	0.3382
DDF	0.0808	0.0140	0.1077	0.0352	0.1435	0.0641	0.1859	0.0982
HDF	0.0767	0.0118	0.0990	0.0279	0.1284	0.0484	0.1610	0.0695
Alg3	0.0042	0.0005	0.0331	0.0014	0.0402	0.0025	0.0481	0.0041

model (1)(3) to contaminate the three original test images by random-valued impulse noise. In the corruption, different noise levels ranging from 5% to 35% with increments of 10% are carried out. Tables 4, 5 and 6 show the values of NCD and NMSE for the restored images “Lena”, “Mandrill” and “Lake”, respectively. Where “Alg5” denotes our proposed algorithm, Algorithm 5. In Fig. 3, we show the restored image by applying VMF, AVMF and Alg5 to the 25% noise corrupted Lena image.

Table 4. Values of NCD and NMSE obtained by different algorithms applied to random-valued impulse noise corrupted Lena image.

Noise level	5%		15%		25%		35%	
Algorithm	NCD	NMSE	NCD	NMSE	NCD	NMSE	NCD	NMSE
VMF	0.0303	0.0016	0.0341	0.0021	0.0398	0.0034	0.0511	0.0071
AVMF	0.0066	0.0005	0.0111	0.0017	0.0254	0.0070	0.0573	0.0213
MAVMF	0.0274	0.0015	0.0325	0.0021	0.0390	0.0034	0.0508	0.0071
QVMF	0.0034	0.0006	0.0106	0.0017	0.0213	0.0035	0.0379	0.0074
VDF	0.0342	0.0021	0.0381	0.0030	0.0457	0.0060	0.0637	0.0152
AVDF	0.0334	0.0016	0.0371	0.0021	0.0453	0.0036	0.0636	0.0080
DDF	0.0322	0.0018	0.0355	0.0024	0.0413	0.0039	0.0540	0.0089
HDF	0.0316	0.0016	0.0355	0.0021	0.0422	0.0035	0.0566	0.0077
Alg5	0.0028	0.0003	0.0085	0.0010	0.0142	0.0016	0.0205	0.0024

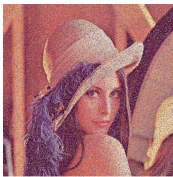
From the Tables 4, 5 and 6 and Fig. 3, we see that Algorithm 5, has lowest NCD and NMSE values and good subjective performance. Moreover, our algorithm is easy to implement. Based on the above results, we conclude that our

Table 5. Values of NCD and NMSE obtained by different algorithms applied to random-valued impulse noise corrupted Lena image.

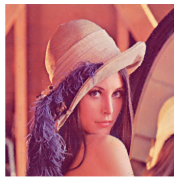
Noise level	5%		15%		25%		35%	
Algorithm	NCD	NMSE	NCD	NMSE	NCD	NMSE	NCD	NMSE
VMF	0.1170	0.0226	0.1230	0.0242	0.1285	0.0266	0.1371	0.0302
AVMF	0.0331	0.0087	0.0580	0.0112	0.0693	0.0164	0.0934	0.0260
MAVMF	0.1159	0.0226	0.1218	0.0242	0.1284	0.0266	0.1371	0.0302
QVMF	0.0578	0.0135	0.0811	0.0276	0.1051	0.0421	0.1297	0.0569
VDF	0.1571	0.0483	0.1698	0.0630	0.1867	0.0825	0.2138	0.1135
AVDF	0.1295	0.0227	0.1408	0.0281	0.1554	0.0364	0.1779	0.0499
DDF	0.1273	0.0292	0.1328	0.0318	0.1411	0.0367	0.1544	0.0459
HDF	0.1183	0.0218	0.1244	0.0235	0.1332	0.0261	0.1438	0.0302
Alg5	0.0161	0.0053	0.0522	0.0104	0.0596	0.0176	0.0818	0.0228

Table 6. Values of NCD and NMSE obtained by different algorithms applied to random-valued impulse noise corrupted Lena image.

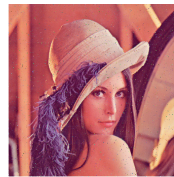
Noise level	5%		15%		25%		35%	
Algorithm	NCD	NMSE	NCD	NMSE	NCD	NMSE	NCD	NMSE
VMF	0.0613	0.0052	0.0705	0.0064	0.0832	0.0090	0.0973	0.0154
AVMF	0.0334	0.0015	0.0405	0.0039	0.0598	0.0123	0.0971	0.0318
MAVMF	0.0604	0.0052	0.0701	0.0064	0.0830	0.0090	0.0972	0.0154
QVMF	0.0360	0.0028	0.0478	0.0056	0.0633	0.0093	0.0854	0.0160
VDF	0.0776	0.0083	0.0826	0.0106	0.0924	0.0165	0.1116	0.0298
AVDF	0.0639	0.0048	0.0713	0.0062	0.0819	0.0096	0.1026	0.0170
DDF	0.0636	0.0063	0.0730	0.0075	0.0822	0.0103	0.0975	0.0167
HDF	0.0611	0.0051	0.0710	0.0063	0.0835	0.0088	0.0996	0.0146
Alg5	0.0282	0.0010	0.0371	0.0028	0.0459	0.0061	0.0566	0.0087



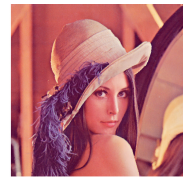
(a)



(b)



(c)



(d)

Fig. 3. Restored images by using different methods. (a) 25% random-valued impulse noise corrupted Lena image, (b) VMF, (c) AVMF, (d) Alg5.

proposed algorithm, Algorithm 5, is competitive with the most of other existing algorithms in removing random-valued impulse noise.

5 Conclusions

We present two classes of effective approaches for removing impulse noise from color images. Our proposed algorithms are effective and performance good. Experimental results show that our proposed algorithms have improvements in NCD and NMSE over most of the existing algorithms.

References

1. Astola, J., Haavisto, P., Neuvo, Y.: Vector median filters. *Proc. IEEE* **78**, 678–689 (1990)
2. Astola, J., Kuosmanen, P.: *Fundamentals of Nonlinear Digital Filtering*. CRC, Boca Raton (1997)
3. Chan, R.H., Ho, C.W., Nikolova, M.: Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Trans. Image Process.* **14**, 1479–1485 (2005)
4. Cheng, L., Hou, Z.-G., Tan, M.: Relaxation labeling using an improved hopfield neural network. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) *Intelligent Computing in Signal Processing and Pattern Recognition. LNCIS*, vol. 345, pp. 430–439. Springer, Heidelberg (2006)
5. Dong, Y., Chan, R.H., Xu, S.: A detection statistic for random-valued impulse noise. *IEEE Trans. Image Process.* **16**, 1112–1120 (2007)
6. Gonzalez, R., Woods, R.: *Digital Image Processing*, 3rd edn. Addison-Wesley, Boston (1993)
7. Gabbouj, M., Cheickh, F.A.: Vector median-vector directional hybrid filter for color image restoration. In: *Proceedings of EUSIPCO-96*, pp. 879–881 (1996)
8. Jin, L., Li, D.: An efficient color-impulse detector and its application to color images. *IEEE Signal Process. Lett.* **14**, 397–400 (2007)
9. Khryashchev, V., Kuykin, D., Studenova, A.: Vector median filter with directional detector for color image denoising. In: *Proceedings of the World Congress on Engineering (London)*, vol. 2 (2011)
10. Karakos, D.G., Trahanias, P.E.: Generalized multichannel image-filtering structures. *IEEE Trans. Image Process.* **6**, 1038–1045 (1997)
11. Lukac, R.: Adaptive vector median filtering. *Pattern Recognit. Lett.* **24**, 1889–1899 (2003)
12. Lukac, R., Plataniotis, K.N., Venetsanopoulos, A.N., Smolka, B.: A statistically-switched adaptive vector median filter. *J. Intell. Robot. Syst. Theory Appl.* **42**, 361–391 (2005)
13. Lukac, R., Smolk, B., Plataniotis, K.N.: Sharpening vector median filters. *Signal Process.* **87**, 2085–2099 (2007)
14. Lukac, R., Smolka, B., Martin, K., Plataniotis, K.N., Venetsanopoulos, A.N.: Vector filtering for color imaging. *IEEE Signal Process. Mag.* **22**, 74–86 (2005)
15. Plataniotis, K.N., Androutsos, D., Venetsanopoulos, A.N.: Color image processing using adaptive vector directional filters. *IEEE Trans. Circ. Syst.* **II(45)**, 1414–1419 (1998)

16. Plataniotis, K.N., Venetsanopoulos, A.N.: Color Image Processing and Applications. Springer, Berlin (2000)
17. Trahanias, P.E., Karakos, D.G., Venetsanopoulos, A.N.: Directional processing of color images: theory and experimental results. *IEEE Trans. Image Process.* **5**, 868–880 (1996)