TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.1, January 2014, pp. 565 ~ 574
DOI: http://dx.doi.org/10.11591/telkomnika.v12i1.3054

A Noise Removal Algorithm of Color Image

Wang Jianwei
College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China
e-mail: jwwang2007@163.com

Abstract

An algorithm of the color image noise removal algorithm is put forward based on the pixel operations. The idea of the algorithm is to read every pixel in a set order and determine whether the pixel level is consistent with the probability density function of impulse noise or not. If it is similar to noise pixel, the number of impulse noise in a certain mask is counted. If the number is less than the given threshold, the pixel is considered as possible noise. The pixel value is not unchanged. Otherwise it is considered as noise, the result mask operation of the pixel is to replace the pixel value. Otherwise it isn't considered as noise, the pixel value is also unchanged. The experimental results show that the algorithm is applicable to the gray noisy image and the color noisy image.It has the advantages of higher speed and more stable noise removal effects.

Keywords: *Color image; Pixel processing; Noise removal; Color model*

1. Introduction

The noise removal problem of the color image is an important research problem in the field of image processing [1]. And it is also applied in different fields [2-5]. The classification principles of typical noise removal methods have three kinds of processing domain, the degree of dependence on the gray level image noise removal and color model.It is divided into the processing based on spatial domain and based on frequency domain according to the domain types. It is also divided into the processing based on a pixel and based on a mask according to the involved pixel number of the spatial domain. The noise removal processing based on a mask may lose the details of the image inevitably [6-8].

According to the degree of dependence on the gray level image noise removal,it is divided into the two kinds. One is to transform the color image to the gray level image, remove the noise and transform the noise removal gray image to the color image. This method may lose some colors during the two transformation courses. The other is to decompose the noisy image to the three components, remove the noise and compose the noise removal components to the color image. So this method needs an uncertain period of time of decomposing and composing operations [9-12].

According to the component correlation of the color model during the course of the color image noise removal, it is divided into the two kinds. One is to transform one color model to another color model with three components with no correlation. This method only removes the noise of the intensity component. And it has the shortcoming of distorted color types. The other is based on the color model with the three correlated components [13]. Each component must be removed the noise respectively. So this method has the advantages of not losing the details and some kinds of colors.

According to the above analysis, the kernel problem of the color image removal noise is researched in this paper. The algorithm based on pixel processing is proposed. And it can remove the impulse noise effectively of the gray level image and the color image based on RGB model.

Received April 29, 2013; Revised July 11, 2013; Accepted August 6, 2013

2. Research Basis

2.1. Analysis of Impulse Noise

Impulse noise is the only one of the visual noise type. And the visual effect of image is directly determined by the effect of noise removal. Impulse noise probability density function can be written as:

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{otherwise} \end{cases}$$

Impulse noise isn't zero when it is Pa or Pb. And it is possible positive or possible negative. If b>a, the gray level b is the luminous point, otherwise the gray level b is the dark point. Impulse noise is normally digitized to the maximum or minimum value. So negative impulse appears in the form of black point (pepper point) in the image and positive impulse appears in the form of white point (salt point) in the image [6, 13].

2.2 Common Image Quality Evaluation Criteria of Noise Removal

How to evaluate the image quality of noise removal is very important. Usually some pixel values of the noisy image are different from the ideal image [3, 13]. Let $f(i,j)$ (i=1,2,…,N, j=1,2,…,M) be the ideal gray image, $g(i,j)$ (i=1,2,…,N, j=1,2,…,M) be the gray image of noise removal. The difference is can be written as: $e = f(i,j) - g(i,j)$

The total difference is can be written as:

$$E = \sum_{i=1}^{N}\sum_{j=1}^{M}[g(i,j) - f(i,j)]$$

The formula for the mean square error can be written as:

$$MSE = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}[f(i,j) - g(i,j)]^2$$

The mean square signal to noise ratio can be written as:

$$MSNR = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}[f(i,j)]^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}[f(i,j) - g(i,j)]^2}$$

Then the evaluation criteria the color image quality of noise removal is got from modifying the above formulae. In this paper, the color image component is considered as the gray image. For example, the ideal color image CGN based on RGB model has component $R1$, component $G1$, and component $B1$. The color image CG of noise removal based on RGB model has component $R2$, component $G2$, and component $B2$. Let the size of the ideal gray image be $N \times M$(i=1,2,…,N, j=1,2,…,M).

The difference is can be written as:

$$e_{RGB} = |R1(i,j) - R2(i,j)| + |G1(i,j) - G2(i,j)| + |R1(i,j) - R2(i,j)| + |B1(i,j) - B2(i,j)|$$

The total difference is can be written as:

$$E_{RGB} = \sum_{i=1}^{N}\sum_{j=1}^{M}[\,|R1(i,j) - R2(i,j)| + |G1(i,j) - G2(i,j)| + |B1(i,j) - B2(i,j)|\,]$$

The formula for the mean square error can be written as:

$$MSE_{RGB} = \frac{1}{3NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\{[R1(i,j)-R2(i,j)]^2+[G1(i,j)-G2(i,j)]^2+[B1(i,j)-B2(i,j)]^2\}$$

The component mean square signal to noise ratio can be written as:

$$MSNR_R = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}[R1(i,j)]^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}[R1(i,j)-R2(i,j)]^2}$$

$$MSNR_G = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}[G1(i,j)]^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}[G1(i,j)-G2(i,j)]^2}$$

$$MSNR_B = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}[B1(i,j)]^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}[B1(i,j)-B2(i,j)]^2}$$

The mean square signal to noise ratio can be written as:

$$MSNR_{RGB} = (MSNR_R + MSNR_G + MSNR_B)/3$$

2.3. Classical Spatial Filters

Classical spatial filters can be applied to gray level image noise removal. The key of the effect of noise removal lies in the mathematical model of the applied spatial filter. The spatial filter types are commonly used as follows [14].

Let *g*(*i*,*j*) (*i*=1,2,…,*N*, *j*=1,2,…,*M*) be the gray image of noise removal, *gn*(i,j) (i=1,2,…, *N*, *j*=1,2,…, *M*) be the gray noisy image. Sij be a rectangular image window with the size of *m*×*n* and the center pixel of (*i*,*j*), then the arithmetic mean filter can be written as:

$$g(i,j) = \frac{1}{mn}\sum_{(s,t)\in S_{ij}} g_n(s,t)$$

The geometric mean filter can be written as:

$$g(i,j) = [\prod_{(s,t)\in S_{ij}} g_n(s,t)]^{1/mn}$$

The median filter can be written as:

$$g(i,j) = \underset{(s,t)\in S_{ij}}{\text{median}}\{g_n(s,t)\}$$

The maximum filter can be written as:

$$g(i,j) = \max_{(s,t)\in S_{ij}}\{g_n(s,t)\}$$

The minimum value filter can be written as:

$$g(i,j) = \min_{(s,t)\in S_{ij}}\{g_n(s,t)\}$$

Then the color image spatial filters of noise removal are got from modifying the above formulae. In this paper, the color image component is considered as the gray image. For example, the color noisy image *CGN* based on RGB model has component *Rn*, component *Gn*, and component *Bn*. The color image *CG* of noise removal based on RGB model has component *R*, component *G*, and component *B* [15]. S_{ij} be a rectangular image window with the size of *m*×*n* and the center pixel of (*i*,*j*), then the arithmetic mean filter can be written as:

$$R(i,j)=\frac{1}{mn}\sum_{(s,t)\in S_{ij}}R_n(s,t)$$

$$G(i,j)=\frac{1}{mn}\sum_{(s,t)\in S_{ij}}G_n(s,t)$$

$$B(i,j)=\frac{1}{mn}\sum_{(s,t)\in S_{ij}}B_n(s,t)$$

And so, in like manner, the other filters are got.

3. Noise Removal Algorithm Based on Pixel Processing

3.1. Algorithm Principle

The prerequisite of noise removal algorithm based on pixel processing is the maximum or minimum value of the impulse noise.The core is to determine the every pixel value of the gray noisy image the maximum or minimum value.

The procedure is to read every pixel in a set order and determine whether the pixel values are consistent with the probability density function of impulse noise or not because the digital impulse noise is the maximum or minimum random value. If it is similar to the noise pixel, then the number of the impulse noise in a certain shape mask is counted. If the number is less than the given threshold, then the pixel is considered as possible noise. The result of the mask operation is to replace the pixel value. Otherwise it is considered as noise, the result of the mask operation without the pixel is to replace the pixel value. Otherwise it isn't considered as noise, the pixel value is unchanged.

3.2. Gray Image Noise Removal Algorithm Based on Pixel Operation

When the pixel value in gray image is represented as 8 bit, the noise is 255 or 0. According to the principle of the algorithm, the algorithm in this paper applied to gray noisy image is given as follows.

Algorithm *g* = gray_NOISE_REMOVAL_PIXEL (*gn*)
Input: gray noisy image *gn*(*i*,*j*) (*i*=1,2,…, *N*, *j*=1,2,…, *M*)
Output: gray image after removing the noise *g*(*i*,*j*) (*i*=1,2,…, *N*, *j*=1,2,…, *M*)
Step 1: Input gray noisy image.
Step 2: Go to Step 8, if the pixels are all disposed.
Step 3: Read the gray level *gn*(*i*,*j*) of the pixel (*i*,*j*) (*i*=1,2,…, *N*, *j*=1,2,…, *M*) in a set order.
Step 4: Go to Step 7 if *gn*(*i*,*j*) isn't the maximum or minimum value.
Step 5: Count the number *c* of the maximum or minimum value in the neighbourhood of (*i*,*j*).
Step 6: Replace the pixel value *gn* (*i*,j)with output *g*(*i*,*j*)of the spatial filter if $c\geq TH$, then go to Step 2.
Step 7: Keep *gn*(*i*,*j*) unchanged, then go to Step 2.
Step 8: Output the result image.

In Step 6, *TH* is an adaptive threshold. Compared with the spatial filters in Section 2.3, there is nearly no mask operations in the algorithm. It is obvious that its execution time can be shorter than the spatial filters.

3.3. Color Image Noise Removal Algorithm Based on Pixel Operation

In this paper, the color image noise removal algorithm based on pixel operation realizes through Algorithm gray_NOISE_REMOVAL_PIXEL call from one time to three times. Namely, the color image component is considered as the gray image. According to the algorithm principle in Section 3.1, the algorithm in this paper applied to color noisy image is given as follows.

Algorithm *CG* = NOISE_REMOVAL_PIXEL (*CGN*)
Input: color noisy image *CGN*(*i*,*j*,*k*) (*i*=1,2,…, *N*, *j*=1,2,…, *M*,*k*=1,2,3)
Output: color image after removing the noise *CG*(*i*,*j*,*k*) (*i*=1,2,…, *N*, *j*=1,2,…, *M*, *k*=1,2,3)
Step 1: Input color noisy image *CGN*.
Step 2: Decompose *CGN* to component *Rn*, component *Gn* and component *Bn*.

Step 3: R = gray_NOISE_REMOVAL_PIXEL (*Rn*).

Step 4: Calculate the component mean square signal to noise ratio *MSNRR*.

Step 5: Compose *R*, *Gn* and *Bn* to the result image *CG* and go to Step 12 if $MSNRR \geq th_{MSNR}$.

Step 6: G = gray_NOISE_REMOVAL_PIXEL (*Gn*).

Step 7: Calculate the component mean square signal to noise ratio *MSNRG.*

Step 8: Compose *R*, *G* and *Bn* to the result image *CG* and go to Step 12 if $(MSNRR + MSNRG)/2 \geq th_{MSNR}$.

Step 9: B = gray_NOISE_REMOVAL_PIXEL (*Bn*) .

Step 10: Calculate the component mean square signal to noise ratio *MSNRB and MSNR*.

Step 11: Compose *R*, *G* and *B* to the result image *CG*.

Step 12: Output the result image *CG*.

In Step 5 and Step 8, th_{MSNR} is an adaptive threshold. Compared with classical color processing, there is no transform from RGB model to another color model in the algorithm. This makes the noise removal time shorter. When there is impulse noise of very low intensity in the color image, only one noisy component has to be processed. With the increase of the noise intensity, the two noisy components should be handled. So the threshold th_{MSNR} is set in the algorithm. It is obvious that its execution time also depends the noise intensity.

4. Experimental Results

4.1. Experimental Results on Gray Noisy Image

The experiments are performed on a Pentium 4 2.4GHZ PC running Windows XP, with 2.0GB memory and one 120 GB hard disk. The below results are got in MATLAB 7.0 environment through editing M files and user defined functions.

Example 1. The idea image in this section is a gray image with 525×640 pixels and 256 gray levels. The noise removal images with different noisy intensity are compared with the ideal image through five spatial filters in Section 2.2 and the algorithm in this paper. The result images are shown in Figure 1, Figure 2, Figure3 and Figure 4. The result images of the noisy image with the intensity of 0.002 through five spatial filters in Section 2.2 and the algorithm in this paper are shown in Figure 1. The noise intensity is 0.02 in Figure 2. The noise intensity is 0.2 in Figure 3. The noise intensity is 0.4 in Figure 4. The performance is compared with execution time, mean square error and mean square signal-to-noise ratio. The corresponding data are shown in shown in Table 1, Table 2, Table 3 and Table 4.

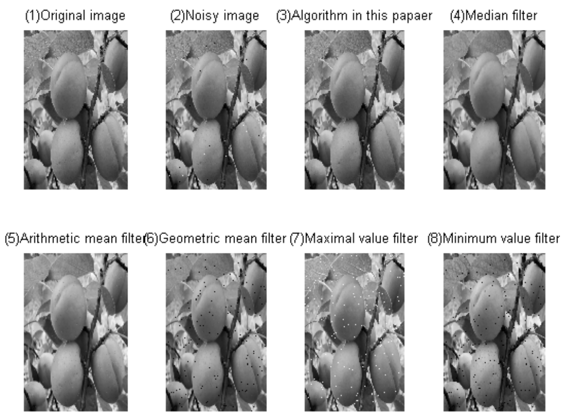

Figure 1. Results through the algorithm and spatial filters of gray noise image (0.002)

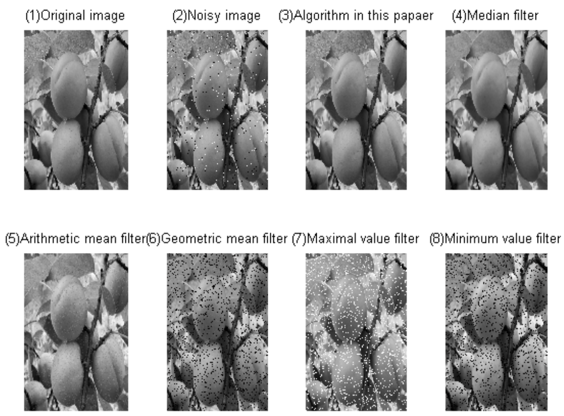

Figure 2. Results through the algorithm and spatial filters of gray noise image (0.02)

Table 1. Data table through the algorithm and classical spatial filters of gray noise image (0.002)

parameter	the algorithm in this paper	median filter	arithmetic mean filter	geometric mean filter	maximum filter	minimum filter
time	2.6720	0.0630	0.0470	0.6880	0.0470	0.0470
MSE	0.0432	10.8562	1.3445e+004	6.4245e+007	188	25153516
MSNR	5.8192e+003	23.4026	4.0484	1.3929	1.8774e+005	3.3917

Table 2. Data table through the algorithm and classical spatial filters of gray noise image (0.02)

parameter	the algorithm in this paper	median filter	arithmetic mean filter	geometric mean filter	maximum filter	minimum filter
time	3.1560	0.0780	0.0470	0.6570	0.0620	0.0620
MSE	0.3753	11.0754	2.1923e+004	5.1533e+008	3065	29897985
MSNR	679.8302	22.9178	2.4671	0.1665	2.4584e+004	2.8571

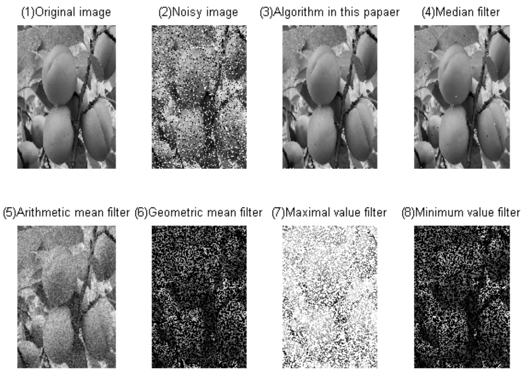

Figure 3. Results through the algorithm and spatial filters of gray noise image (0.2)

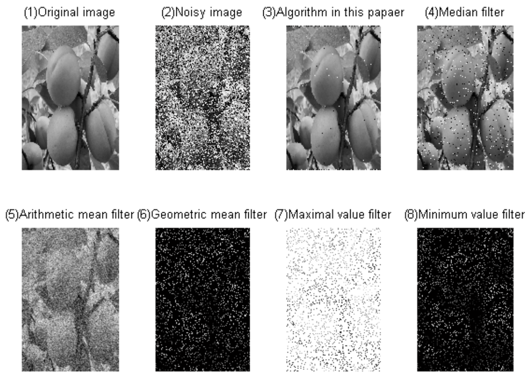

Figure 4. Results through the algorithm and spatial filters of gray noise image (0.4)

Table 3. Data table through the algorithm and classical spatial filters of gray noise image (0.2)

parameter	the algorithm in this paper	median filter	arithmetic mean filter	geometric mean filter	maximum filter	minimum filter
time	8.0160	0.0630	0.0470	0.5780	0.0630	0.0470
MSE	4.2291	14.1498	1.2043e+005	3.4607e+009	20670	61271684
MSNR	59.8847	17.8147	0.4438	0.0245	4.0791e+003	1.3862

Table 4. Data table through the algorithm and classical spatial filters of gray noise image (0.4)

parameter	the algorithm in this paper	median filter	arithmetic mean filter	geometric mean filter	maximum filter	minimum filter
time	13.1100	0.0630	0.0310	0.5000	0.0470	0.0470
MSE	9.7107	21.8618	2.6498e+005	4.8733e+009	21130	76720008
MSNR	26.3543	11.6795	0.2065	0.0175	3.5898e+003	1.1115

From the result images and the data, the algorithm is better than the other methods with the increase of the noise intensity.

4.2. Experimental Results on Color Noisy Image Based on RGB Color Model

The experiment results in this section are got by Algorithm NOISE_REMOVAL_PIXEL.

Example 2. The ideal image in this section is a color image with 525×640 pixels and 256^3 kinds of colors. The noise removal images with different noisy intensity are compared with the ideal image through and the algorithm in Section 3.3. The result images are shown in Figure 5, Figure 6, Figure 7 and Figure 8. The result images of only dealing with component *R*, component *G*, component *B* and three components with the intensity of 0.002 through the algorithm in this paper are shown in Figure 5. The noise intensity is 0.02 in Figure 6. The noise intensity is 0.2 in Figure 7. The noise intensity is 0.4 in Figure 8. The performance is compared with execution time, mean square error and mean square signal-to-noise ratio. The corresponding data are shown in shown in Table 5, Table 6, Table 7 and Table 8.

Figure 5. Results through the algorithm in this paper of color noise image (0.002)

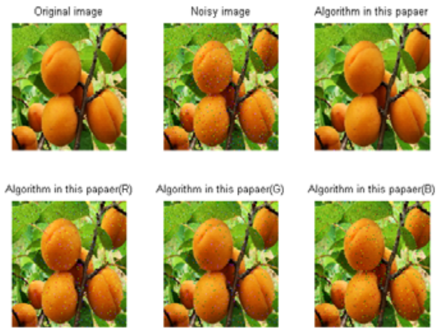

Figure 6. Results through the algorithm in this paper of color noise image (0.02)

Table 5. Data table through the algorithm in this paper of color noise image (0.002)

parameter	three components	Component R	Component G	Component B
time	9.5620	2.6560	2.6090	4.2970
MSE	0.0553	0.1690	0.1422	0.1686
MSNR	4.4207e+003	1.6684e+003	2.6520e+003	2.2223e+003

Table 6. Data table through the algorithm in this paper of color noise image (0.02)

parameter	three components	Component R	Component G	Component B
time	11.0610	3.1870	3.0930	4.7810
MSE	0.4481	1.5669	1.5559	1.8414
MSNR	503.2934	236.8608	268.7617	197.3592

Figure 7. Results through the algorithm in this paper of color noise image (0.2)

Figure 8. Results through the algorithm in this paper of color noise image (0.4)

Table 7. Data table through the algorithm in this paper of color noise image (0.2)

parameter	three components	Component R	Component G	Component B
time	25.5450	8.0150	7.9210	9.6090
MSE	4.5627	15.6368	15.5544	18.4994
MSNR	49.4118	24.5453	25.8761	18.9766

Table 8. Data table through the algorithm in this paper of color noise image (0.4)

parameter	three components	Component R	Component G	Component B
time	40.8440	13.4530	13.2500	14.1410
MSE	10.3294	31.7394	31.5016	37.1585
MSNR	21.7814	11.1466	11.7508	8.8865

From the result images and the data, the component result is considered as the last result when the noise intensity is lower than 0.1. The two component results may be chosen to be the last result when the noise intensity is in the range of [0.1 0.2].

Example 3 The ideal image in this section is a color image with 535×640 pixels and 256^3 kinds of colors. The noise removal images with different noisy intensity are compared with the ideal image through some spatial filters in Section 2.2 and the algorithm in this paper. The result images are shown in Figure 9, Figure 10, Figure 11 and Figure 12. The result images of the noisy image with the intensity of 0.002 through some spatial filters in Section 2.2 and the algorithm in this paper are shown in Figure 9. The noise intensity is 0.02 in Figure 10. The noise intensity is 0.2 in Figure 11. The noise intensity is 0.4 in Figure 12. The performance is compared with execution time, mean square error and mean square signal-to-noise ratio. The corresponding data are shown in shown in Table 9, Table 10, Table 11 and Table 12.

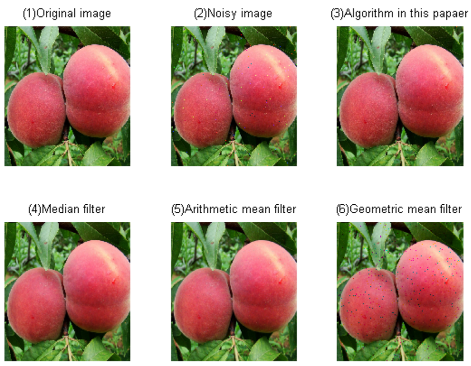

Figure 9. Results through the algorithm and spatial filters of color noise mage (0.002)

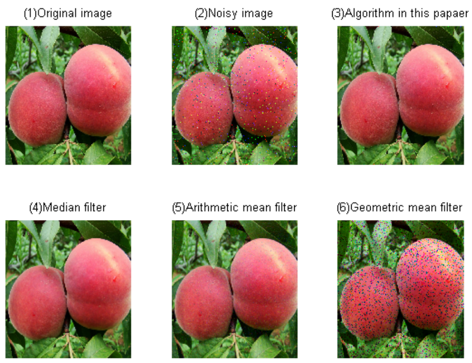

Figure10. Results through the algorithm and spatial filters of color noise mage (0.02)

Table 9. Data table through the algorithm and classical spatial filters of color noise image (0.002)

parameter	the algorithm in this paper	median filter	arithmetic mean filter	geometric mean filter
time	10.1100	0.1720	0.2500	2.0930
MSE	0.1398	12.7134	54.0086	6.0603e+007
MSNR	3.3151e+003	18.8931	4.4388	1.4684

Table 10. Data table through the algorithm and classical spatial filters of color noise image (0.02)

parameter	the algorithm in this paper	median filter	arithmetic mean filter	geometric mean filter
Time(s)	11.6720	0.1720	0.2500	2.1090
MSE	0.5898	13.0169	99.2984	4.5544e+008
MSNR	420.3624	18.4512	2.4124	0.2164

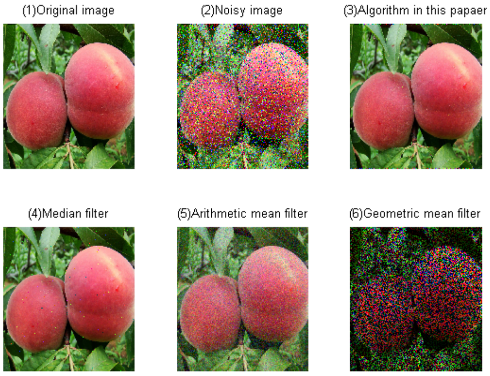

Figure 11. Results through the algorithm and spatial filters of color noise mage (0.2)

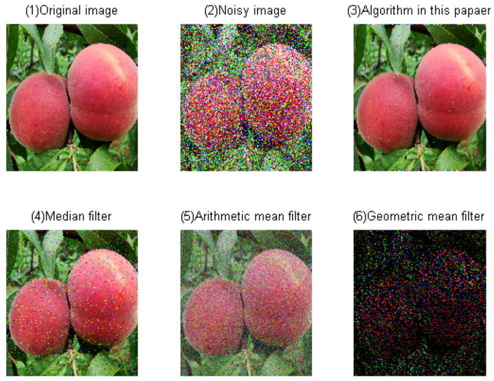

Figure 12. Results through the algorithm and spatial filters of color noise mage (0.4)

Table 11. Data table through the algorithm and classical spatial filters of color noise image (0.2)

parameter	the algorithm in this paper	median filter	arithmetic mean filter	geometric mean filter
time	26.5150	0.1870	0.2820	1.7340
MSE	5.4255	16.5065	712.9832	3.1275e+009
MSNR	44.1984	14.5379	0.3401	0.0320

Table 12. Data table through the algorithm and classical spatial filters of color noise image (0.4)

parameter	the algorithm in this paper	median filter	arithmetic mean filter	geometric mean filter
time	43.0480	0.1870	0.2500	1.5940
MSE	12.2334	24.8358	1.7201e+003	4.3935e+009
MSNR	19.6070	9.6500	0.1420	0.0227

From the result images and the data, the algorithm is better than the other methods with the increase of the noise intensity.

5. Conclusion

The impulse noise removal algorithm based on pixel processing is given. The algorithm not only can remove the noise in the gray image, but also can remove the noise in the color image based on three correlative components by call the algorithm from one time to three times. Comparing with the other spatial filters, the method keeps the more details and has higher mean square signal to noise ratio.

Acknowledgment

Supported by Scientific Research Fund of Heilongjiang Provincial Education Department (No. 12521101, No.12533020)

References

[1] Arjan Gijsenij, Theo Gevers, Joost van de Weijer. Computational Color Constancy: Survey and Experiments. *IEEE Transactions on Image Processing*. 2011; 20(9): 2475-2489.

[2] Tzu-Chao Lin, Pao-Ta Yu. Salt-Pepper Impulse Noise Detection and Removal Using Multiple Thresholds for Image Restoration. *Journal of Information Science and Engineering*. 2006; 22: 189-198.

[3] Jamal Ibaa, Usman Akram M, Tariq Anam. Retinal image preprocessing: Background and noise segmentation. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(3): 537-544.

[4] Hartati Sri, Harjoko Agus, Supardi Tri Wahyu. The digital microscope and its image processing utility. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(3): 565-574.

[5] Wang Kai-Ping, Liu Chiang-Lung, Lou, Der-Chyuan. Meaningful color image sharing without pixel expansion problem. *International Journal of Advancements in Computing Technology*. 2012; 4(20): 525-535.
[6] Knig Maxime, Cavalieri André VG, Jordan Peter, Delville Joël, Gervais Yves, Papamoschou Dimitri. Farfield filtering and source imaging of subsonic jet noise. *Journal of Sound and Vibration*. 2013; 332(18): 4067-4088.
[7] Setyaningsih Emy, Iswahyudi Catur, Widyastuti Naniek.Image encryption on mobile phone using super encryption algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(4): 835-843.
[8] Nasri Mohsen, Helali Abdelhamid, Sghaier Halim, Maaref Hassen. Efficient JPEG 2000 image compression scheme for multihop wireless networks. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(2): 311-318.
[9] Sastra Nyoman Putra, Wirawan, Hendrantoro Gamantyo. Virtual view image over wireless visual sensor network.*Telkomnika*. 2011; 9(3): 483-488.
[10] Muntasa Arif, Sirajudin Indah Agustien, Purnomo Mauridhi Hery. Appearance global and local structure fusion for face image recognition. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(1): 125-132.
[11] Borja Rodríguez-Cuenca, Jose A. Malpica, Maria C. Alonso. A Spatial Contextual Postclassification Method for Preserving Linear Objects in Multispectral Imagery. *IEEE Transactions on Geoscience and Remote Sensing*. 2013; 51(1): 174-183.
[12] http://en.wikipedia.org/wiki/RGB_color_spaces 2013.
[13] Rafael C Gonzalez, Richard E Woods. Digital Image Processing, 2ed. Beijing: Publishing House of Electronics Industry. 2009.
[14] Leah Bar, Nir Sochen, and Nahum Kiryati. Image Deblurring in the Presence of Salt-and-Pepper Noise. *Lecture Notes in Computer Science*. 2005; 3459: 107-118.
[15] Yu Qian, Kang Zhang, Fang Qiu. *Spatial Contextual Noise Removal for Post Classification Smoothing of Remotely Sensed Images*. *SAC'05*. 2005.