Iterative Removing Salt and Pepper Noise based on Neighbourhood Information

Liu Chun
College of Computer Science and Information Technology
Daqing Normal University
Daqing, China

Liu Shaohui*
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, China

Sun Bishen
Twenty-seventh Research Institute
China Electronic Technology Group Corporation
Zhengzhou, China

Tan Kun, Ma Yingrui
College of Computer Science and Information Technology
Daqing Normal University
Daqing, China

***Abstract*—Denoising images is a classical problem in low-level computer vision. In this paper, we propose an algorithm which can remove iteratively salt and pepper noise based on neighbourhood while preserving details. First, we compute the probability of different window without free noise pixel by noise ratio, and then determine the size of window. After that the corrupted pixel is replaced by the weighted eight neighbourhood pixels. If the neighbourhood information does not satisfy the de-noising condition, the corrupted pixels will recover in the subsequent iterations.**

Keywords—Salt and pepper noise; noise detection; neighbourhood similarity; detail preserving denoising

I. Introduction

Salt and pepper noise (SPN) usually comes from the image sensor, transmission channel and decoding processing. The corrupted pixels take either maximum or minimum grey value, contributing to black and white dots on image. Most of tasks about computer vision, for example, image segmentation, feature extraction, image recognition, are strongly influenced by impulse noise. Therefore, it is necessary for removing the salt and pepper noise on image.

In the field of SPN reduction, most of the proposed methods are based on two phases method, namely, noise detection and filter. Currently, there are many noise detection and judgment methods, for example, Laplacian convolution [7], maximum gradient difference in window [5], minimum gradient difference in window [3]. For the filter of the second phase removing SPN, many approaches are proposed in recently years. For example, the median filter (MF), the switch median filter [1], the adaptive median filter (AMF) [2], the noise adaptive fuzzy switching median filter (NAFSMF) [3], [6], [8], the adaptive weighted mean filter (AWMF) [4], the using long-range correlation filter (LRC) [9]. Most of them are based on the improvement of traditional median filter or mean value in window. Thus, they will cause inevitably fuzzy edge in image restoration. Moreover, the size of filtering window has remarkable impact on the effect of SPN image filtering. There are many ways to choose size of filtering window, for example, 3×3, 5×5, 7×7, or even if the current filtering window does not have noise-free pixel, the filtering window will be expanded until 9×9. The way of keeping the same size of filtering window is clearly unreasonable. It is obvious that there are no enough clean pixels to restore the corrupted pixels in very high noise ratio window if choosing the window 3×3. However, if choosing a big filtering window (9×9) or much larger one, that means amount of calculation in the low noise image recovering. Thus, how to choosing a most appropriate size of the window is very important for removing SPN in images.

In this paper, we proposed a new algorithm for SPN called the iterative restoration based on neighbourhood information. As we know, if the noise pixel has enough neighbourhood information, we can determine the corrupted pixel which belongs to edge area or flat area approximately, then based on this information, a perfect recovery can be achieved. When eight-neighbourhood information of the destroyed pixel are not complete or even are not available, the recovery needs through good blocks from global of image to assist the process. It will be an iteratively process until the eight-neighbourhood can remove noise pixels. The simple example is shown in Fig. 1. It is worth noting that the aforementioned algorithms may be bad in high or low SPN. But our present pattern has better recovery both in case of high or low SPN. The superiority of our algorithm is very obvious.

The rest of this paper is organized as follows. The Section II will give the details of proposed algorithm. The detection of noise level is first analyzed, and then the removing noise algorithm is proposed based on the neighbourhood pixels. And experiment results are conducted in Section III. The Section IV gives the conclusion.

This work is partially supported by the Science Research Foundation of Daqing Normal University under Grant No.14ZR02.

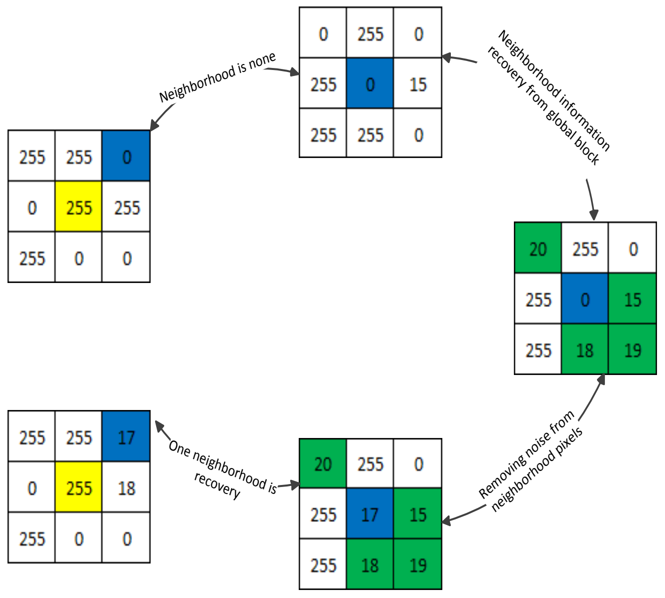

Fig. 1. The noise pixel and neighbourhood information recovery process.

II. Proposed Algorithm

Following, we will illustrate the algorithm from the choosing window, noise detection, iteratively removing noise with neighbourhood pixels.

A. Choosing a Window

Assuming that SPN is random distribution, and the ratio of noise is r, $r = \frac{\#noise\ pixel}{\#total\ pixels}$. r can be acquired from priori or according to the noise density of image. Select an $m \times m$ window, the probability of pixel in window is $P(m) = r^{m\times m}$, $0 \le r < 1$. It means the probability of all pixels are noise in window $m \times m$. For example, if $m = 3$, the probability of no undamaged in window 3× 3 is $r^{3\times3}$. Then we can estimate the relationship among $P(m)$, r and m. The Table I shows the result.

TABLE I. Relationship Among $P(m)$, r and m

m / r(%)	3	5	7	9	11
30	2.0×10^{-5}	8.5×10^{-14}	2.4×10^{-26}	4.4×10^{-43}	5.4×10^{-64}
40	2.6×10^{-4}	1.1×10^{-10}	3.1×10^{-20}	5.8×10^{-33}	7.1×10^{-49}
50	0.0020	3.0×10^{-8}	1.8×10^{-15}	4.1×10^{-25}	3.8×10^{-37}
60	0.010	2.8×10^{-6}	1.3×10^{-11}	1.1×10^{-18}	1.4×10^{-27}
70	0.040	1.3×10^{-4}	2.6×10^{-8}	2.8×10^{-13}	1.8×10^{-19}
80	0.13	0.0038	1.8×10^{-5}	1.4×10^{-8}	1.9×10^{-12}

From the Table II, if selecting 3×3 window, the window contains free noise pixels at least 99.9% when r below 0.3. If selecting 9×9 window, the window contains free noise pixels at least 99.9% when r below 0.7. The formula of chosen window defined as

$$m = \begin{cases} 3 & if\ r < 0.3 \\ 5 & if\ 0.3 \le r < 0.5 \\ 7 & if\ 0.5 \le r < 0.6 \\ 9 & if\ 0.6 \le r < 0.7 \\ 11 & if\ r \ge 0.7 \end{cases} \quad (1)$$

If it is difficult to get value of r, the default value of m set 7.

B. Noise Detection

$X(i,j)$ denotes the pixel value of coordinate (i,j) on image. $N(i,j)$ records (i,j) whether noise-free pixel or not (0 means undamaged pixel, 1 means damaged pixel). For the SPN, how to further determine whether it is real noise becomes intractability. It has two cases shown in Fig. 2.

In the case of (1), there is no undamaged pixel to remove SPN. But in the previous step, we know the probability of containing uncorrupted pixels at least 99.9%. Therefore, it can be sure that the pixels in window are white block or black block. Updating the pixel at location (i,j) using

$$X(i,j) = argmax_{k\in\Omega_{i,j}^{m}}(c(k)) \quad (2)$$

where $c(k)$ is the number of pixel value for k, $\Omega_{i,j}^{m}$ is the pixels in $m \times m$ window which centre of location (i,j).

In the case of (2), we use G_Ω indicating the noise free pixels in window, defined as

$$G_\Omega = \{X(k,l)|X(k,l) \ne 0\ and\ 255, |k-i| \le w, |l-j| \le w\} \quad (3)$$

$w = (m-1)/2$, calculating the mean of not maximum and minimum pixel values and standard deviation (SD) in window.

$$SD = \sqrt[2]{\frac{\sum_{X(k,l)\in G_\Omega}(X(k,l)-mean)^2}{c_{G_\Omega}}} \quad (4)$$

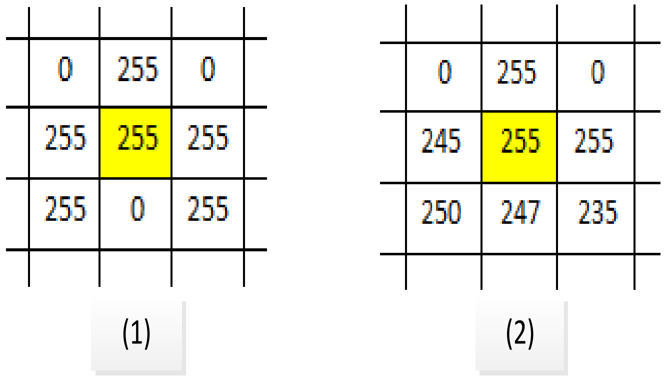

Fig. 2. Two cases of extreme value in window.

where c_{G_Ω} is the number of clean pixels in window. This step only adapt for flat area in image. So the SD must less than a threshold δ. δ which sets to smaller is reasonable, or according to algorithm of OTSU may be more accuracy. In this paper, we simply set $\delta = 10$. The detailed algorithm is as follows:

Noise Detection Algorithm

```
1)  **for** each pix$(i,j)$ in noise image X
2)      **if** $X(i,j) == 0\ or\ X(i,j) == 255$
3)          **if** $\forall X(r,t) \in \Omega_{i,j}^m, X(r,t) == 0\ or\ X(r,t) == 255$
4)              **then** $X(i,j) = argmax_{k\in\Omega_{i,j}^m}(c(k)), N(i,j) = 0$
5)          **else if** $SD < \delta\ \&\ |X(i,j) - mean| < SD$
6)              **then** $N(i,j) = 0$
7)          **else**
8)              $N(i,j) = 1$
9)      **endif**
10)     **else**
11)         $N(i,j) = 0$
12)     **endif**
13) **end for**
```

C. Noise Removing

Assuming that $X(i,j)$ is the noise pixel at location (i,j). Its eight-neighbourhood is defined as

$$\Omega_{\text{EN}} = \{(k,l)|(k,l) \neq (i,j), |k-i| \leq 1, |l-j| \leq 1\} \quad (5)$$

Setting up eight $m \times m$ windows centered as Ω_{EN}. They are respectively compared the $m \times m$ window centered as (i,j), using the similarity between windows and weighted filtering. The similarity comparison is given as

$$KL(\alpha||\beta) = \alpha\, log\frac{\alpha}{\beta} + (1-\alpha)log\frac{1-\alpha}{1-\beta} \quad (6)$$

In this paper, α is the pixels of window centered at (i,j), β is the pixels of window centered as Ω_{EN}. For avoiding overflow and dividing by zero, the image pixels scaling to (0,1) before using formula (6), given as

$$X(i,j) = \frac{X(i,j)+\mu}{256} \quad (7)$$

μ sets a smaller value 0.001, avoiding $X(i,j)$ equal to 0 or 1.

Taking partial derivatives of formula (6)

$$\frac{\partial KL}{\partial \alpha} = \log(\frac{\alpha}{\beta} \times \frac{1-\beta}{1-\alpha}) \qquad \frac{\partial KL}{\partial \beta} = \frac{1-\alpha}{1-\beta} - \frac{\alpha}{\beta} \quad (8)$$

By letting formula (8) equals to zero. Then it can be obtained easily that the minimum value of formula (6) is 0 at $\alpha = \beta$. So the more similar between α and β, the smaller value of $KL(\alpha||\beta)$. If $\beta = 0.3$, the graph shows as Fig. 3.

The similarity between window centered at (i,j) and window centered at (k,l) defined as

$$s(k,l) = \sum_{p=-w}^{w}\sum_{q=-w}^{w}\big(1 - N(i+p,j+q)\big)\big(1 - N(k+p,l+q)\big) \times KL(X(i+p,j+q)||X(k+p,l+q)) \quad (9)$$

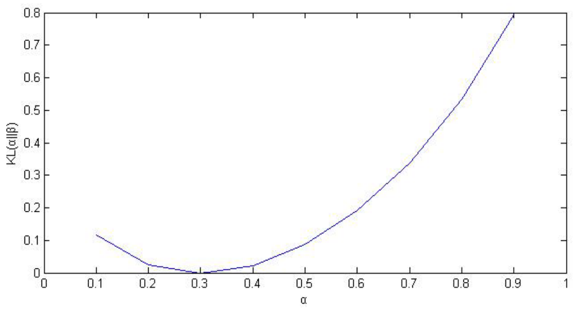

Fig. 3. The relationship between α and β.

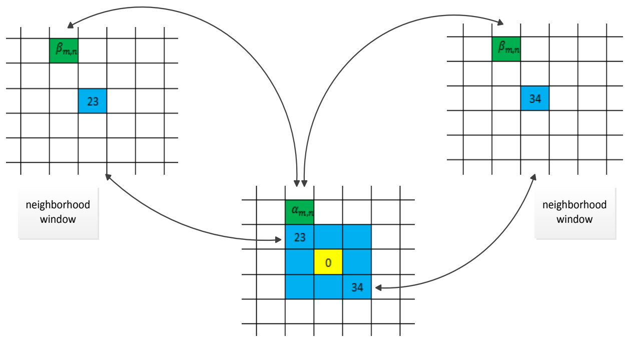

Fig. 4. The similarity comparison and computing.

From this equation, we can observe that the smaller $s(k,l)$, the stronger the correlation and the filtering will apt to use larger weight value. As shown in Fig. 4, illustrates the calculation of similarity between $\Omega_{i,j}^m$ and $\Omega_{k,l}^m$ with same location, where $(k,l) \in \Omega_{\text{EN}}$. It is noted that the calculation of similarity only uses clean pixels in window.

Let $Y(i,j)$ denote filtering pixel value for the image $X(i,j)$, it can be defined as

$$Y(i,j) = \big(1 - N(i,j)\big) \times X(i,j) + N(i,j) \times Z(i,j) \quad (10)$$

where

$$Z(i,j) = \frac{\sum_{k,l\in\Omega_{\text{EN}}}\frac{1}{s(k,l)} \times X(k,l)}{\sum_{k,l\in\Omega_{\text{EN}}}\frac{1}{s(k,l)}} \quad (11)$$

In the formula (9), we could not consider the situation of $s(k,l) = 0$, to illustrate this problem, given formula as

$$comp_{k,l} = \sum_{p=-w}^{w}\sum_{q=-w}^{w}\big(1 - N(i+p,j+q)\big)\big(1 - N(k+p,l+q)\big) \quad (12)$$

It means the total number of undamaged pixels in same location between window centered at (i,j) and window centered at (k,l). If $comp_{k,l} = 0$, we do not compute $s(k,l)$. If $comp_{k,l} \neq 0$, but $s(k,l) = 0$, $s(k,l)$ is set to the value of σ, σ is a very small value of 0.0001. If $\forall(k,l), (k,l) \in \Omega_{\text{EN}}$, $comp_{k,l} \equiv 0$, i.e., the neighbourhood information is not enough for removing noise. The corrupted pixels at location (i,j) will be recovered in the subsequent iterations.

The convergence condition is that any $Y(i,j)$ has not been updated in one iteration or any $Y(i,j)$ does not equal the initial value. If some $Y(i,j)$ equals initial value end of iteration, then the iteration is terminated. And $Y(i,j)$ is set to the mean of clean pixels (G_Ω) in $m \times m$ window centered at (i,j).

Iterative Recovery Algorithm

```
1)  initialize iter = true, flag = true $Y(i,j) = 1$
2)  while iter and flag
3)      iter = false
4)      flag = false
5)      for each pix$(i,j)$ in noise image X
6)          if $N(i,j) == 1$
7)              then for $(k,l)$ in $\Omega_{EN}$
8)                  compute $comp_{k,l}$, $s(k,l)$
9)              if $\forall(k,l),(k,l) \in \Omega_{EN}, comp_{k,l} \equiv 0$
10)                 then iter=true
11)             else
12)                 $Y(i,j) = Z(i,j)\ N(i,j) = 2$
13)                 flag = true
14)             endif
15)         else
16)             $Y(i,j) = X(i,j)$
17)             flag = true
18)     end for
19)     update $N(i,j)$
20)     for each $(i,j)$ in N
21)         if $N(i,j) == 2$
22)             then $N(i,j) = 0, X(i,j) = Y(i,j)$
23)         endif
24)     end for
25) end while
26) if flag == false
27)     for each pix$(i,j)$ in noise image Y
28)         if $Y(i,j) == 1$
29)             $Y(i,j) = mean(G_\Omega)$
30)         endif
31)     end for
32) endif
```

III. Experimental Results

In our experiment tests, five different methods include median filtering (MF), NAFSMF [3], LRC [9], AMF [2], AWMF [4] are used to evaluate the performance of our proposed algorithm. The extensive experiments are conducted to verify the performance. In this paper, only the four tested images "Lena", "boat", "bridge" and "Elaine" are reported. All the image size is 512×512. We use peak signal to noise ratio (PSNR) to evaluate the performance of different methods. The PSNR is defined as

$$PSNR = 10\,log_{10}\frac{255^2}{MSE} \tag{13}$$

$$MSE = \frac{1}{M \times N}\sum_{i=0}^{i=M}\sum_{j=0}^{j=N}(Y(i,j) - M(i,j))^2 \tag{14}$$

where $M(i,j)$ is the pixel value of original image. The Table II gives the results compared with other five different methods. From this table, proposed method achieves the best results in most of cases. Actually, there are only two cases which the proposed algorithm ranks the second position for 4 images with 9 noise levels. And the AWMF algorithm achieves the best results for the Bridge image with 80% and 90% noise levels. The black font in the table means the best results. In most of case, the proposed algorithm achieves the PSNR over 20 db even the noise level is larger than 90%.

Assuming that SPN is random distribution, the ratio between salt and pepper is 1, namely, the number of salt pixels is almost equal to that of pepper pixels. In Fig. 5, the original image with 90% SPN, the whole process of iteration is shown. The each iterated middle-result image with removing noise pixel by fully using neighbourhood information is shown, the progressive quality is rather obvious from the (a) to (f). The experimental results demonstrate that the method possesses good vision effect.

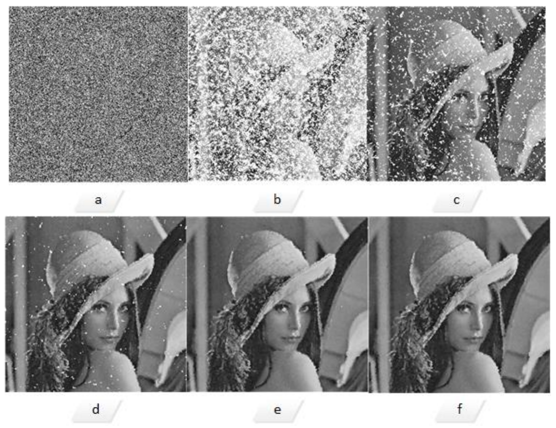

Fig. 5. The iterative restoration for "Lena" with 90% SPN.(a) Original image (b) First iteration (c) Second iteration (d) Third iteration (e) Fourth iteration (f) Convergent image.

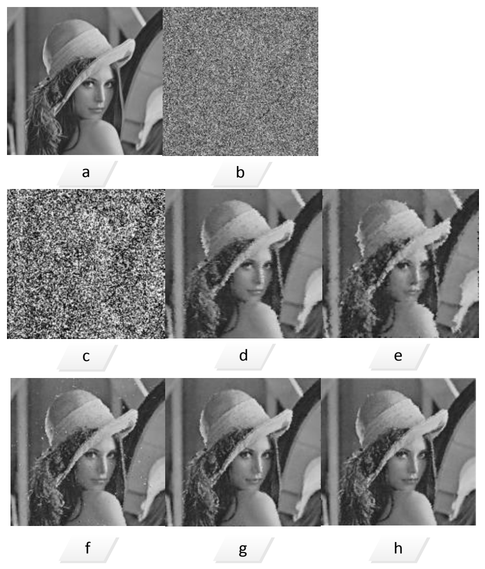

Fig. 6. The restoration for "Lena" with different methods (a) Original image (b) Image with 90% SPN (c) MF (d) LRC (e) AMF (f) NAFSMF (g) AWMF (h) Proposed.

TABLE II. RESULT OF PSNR (DB) FOR DIFFERENT ALGORITHM WITH VARIOUS NOISE LEVEL

Image	**Algorithms**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Lena	MF	32.50	29.49	23.79	19.14	15.32	12.41	10.02	8.15	6.60
	LRC	40.96	36.69	32.99	29.38	26.17	23.81	22.21	21.34	21.18
	NAFSM	41.17	37.87	35.27	33.36	31.72	30.03	28.22	26.18	22.08
	AMF	36.58	34.75	32.69	31.12	29.35	27.91	26.35	24.35	21.73
	AWMF	38.16	36.22	34.99	33.80	32.42	31.03	29.59	27.84	25.37
	Proposed	**43.49**	**39.73**	**37.94**	**35.87**	**34.16**	**32.34**	**30.36**	**28.30**	**25.58**
Boat	MF	29.55	27.14	22.96	18.88	15.17	12.34	10.04	8.14	6.64
	LRC	38.59	34.29	30.08	26.64	23.61	21.42	19.97	19.24	19.00
	NAFSM	38.14	34.62	32.33	30.31	28.73	27.30	25.72	24.02	20.93
	AMF	33.36	31.59	29.69	28.30	26.71	25.33	23.94	22.24	19.95
	AWMF	35.01	33.34	32.05	30.69	29.58	28.17	26.89	25.24	23.01
	Proposed	**40.98**	**36.98**	**35.03**	**32.95**	**31.17**	**29.33**	**27.53**	**25.55**	**23.33**
Bridge	MF	26.05	24.53	21.51	18.03	14.68	11.95	9.68	7.88	6.38
	LRC	33.24	29.34	26.33	23.43	21.09	19.34	18.16	17.48	17.17
	NAFSM	34.57	31.29	28.95	27.25	25.74	24.22	22.86	21.43	18.89
	AMF	30.03	28.74	27.20	25.76	24.43	23.15	21.85	20.36	18.35
	AWMF	32.58	30.82	29.37	28.10	26.88	25.67	24.39	**22.91**	**21.06**
	Proposed	**35.68**	**32.17**	**30.48**	**28.78**	**27.30**	**25.87**	**24.40**	22.81	20.78
Elaine	MF	31.58	28.93	23.75	19.19	15.36	12.45	10.06	8.21	6.68
	LRC	38.64	35.19	32.51	22.19	26.91	24.66	23.00	22.33	22.13
	NAFSM	40.52	37.20	35.16	33.51	31.93	30.57	28.81	26.99	23.02
	AMF	35.06	34.13	32.90	31.56	30.35	28.90	27.46	25.76	23.08
	AWMF	38.80	36.94	35.44	34.06	32.73	31.50	30.18	28.82	26.82
	Proposed	**41.20**	**37.73**	**36.09**	**34.43**	**33.25**	**31.87**	**30.38**	**28.83**	**26.84**

In Fig. 6, we show the PSNR of different methods for "Lena" with 90 percent SPN. Obviously, MF, LRC, AMF and NAFSMF are not efficient for high noise ratio in image. But AWMF is not ideal for low noise ratio in image. The details of filtering results are in Table II.

In Fig. 7, we show the average time of different methods for image filtering. MF, NAFSMF, AWMF keep a low filtering time no matter noise ratio. Although more time than other when noise is high in our algorithm, the PSNR is highest from our filtering. From the trend of curve, our approach is reasonable. Because it is an iterative process, the process time will be longer with high SPN.

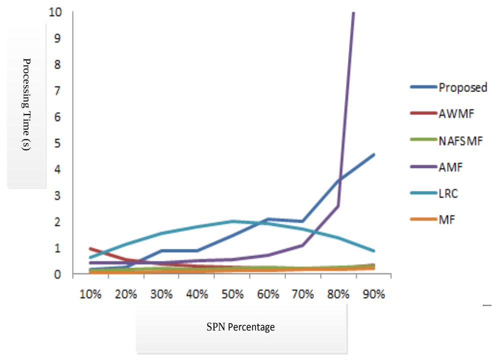

Fig. 7. The average time of different methods for tested images.

IV. CONCLUSIONS

In this paper, a highly efficient denoising method is presented no matter high or low SPN. It is an iterative process by effectively using neighbourhood information and global information. Our present pattern has a better recovery both in case of high or low SPN by experiment. Importantly, our ideal is novel in the field of SPN and obtained good result. In addition, the method can also be used in 3D models by simple extension. The future work includes how to apply this algorithm into the practical application, especially the depth image denoising and 3D points denoising. In addition, how to accelerate the speed is also another research point.

REFERENCE

[1] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," IEEE Trans. Circuits Syst. II: Analog Digital Signal Process., vol. 46, no. 1, pp. 78–80, 1999.

[2] H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithms and results," IEEE Trans. Image Process., vol. 4, no. 4, pp. 499–502,1995.

[3] K. K. V. Toh and N. A.M. Isa, "Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction," IEEE Signal Process. Lett., vol. 17, no. 3, pp. 281–284, 2010.

[4] P. Zhang and F. Li, "A New Adaptive Weighted Mean Filter for Removing Salt-and-Pepper Noise," IEEE Signal Process. Lett., vol. 21, no. 10, pp. 1280–1283, 2014.

[5] C. Lu, T. Chou, "Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter," Pattern Recognition Letters, vol.33,pp. 1287–1295, 2012.

[6] H. Xu, G. Zhu, H. Peng, and D. Wang, "Adaptive fuzzy switching filter for images corrupted by impulse noise," Pattern Recognition Letters, vol.25, pp. 1657-1663, 2004.

[7] S. Tai, S. Yang, "A Fast Method For Image Noise Estimation Using Laplacian Operator and Adaptive Edge Detection," International Symposium on Communications, Control and Signal Processing, Malta, pp. 1077-1081, 2008.

[8] K. K. V. Toh, H. Ibrahim, and M. N. Mahyuddin, "Salt-and-pepper noise detection and reduction using fuzzy switching median filter," IEEE Trans. Consumer Electron., vol. 54, no. 4, pp. 1956–1961, Nov.2008.

[9] Z. Wang and D. Zhang,"Restoration of Impulse Noise Corrupted Images Using Long-Range Correlation" IEEE Signal Process. Lett., vol. 5, no. 1, pp. 4–7, 1998.