

Modified Adaptive Gaussian Filter for Removal of Salt and Pepper Noise

Zuoyong Li¹, Kezong Tang², Yong Cheng³, Xiaobo Chen⁴ and Chongbo Zhou⁵

¹ Department of Computer Science, Minjiang University, Fuzhou 350108, China

² School of Information Engineering, Jingdezhen Ceramic Institute, Jingdezhen 333403, China

³ School of Communication Engineering, Nanjing Institute of Technology, Nanjing 211167, China

⁴ Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China

⁵ School of Physics and Engineering, Qufu Normal University, Qufu 273165, China

[email: fzulzytdq@126.com, tangkezong@126.com, chengyong@njit.edu.cn, xbchen82@gmail.com, zhouchongbo@163.com]

*Corresponding author: Zuoyong Li

Received March 25, 2015; revised June 30, 2015; accepted July 9, 2015;

published August 31, 2015

Abstract

Adaptive Gaussian filter (AGF) is a recently developed switching filter to remove salt and pepper noise. AGF first directly identifies pixels of gray levels 0 and 255 as noise pixels, and then only restored noise pixels using a Gaussian filter with adaptive variance based on the estimated noise density. AGF usually achieves better denoising effect in comparison with other filters. However, AGF still fails to obtain good denoising effect on images with noise-free pixels of gray levels 0 and 255, due to its severe false alarm in its noise detection stage. To alleviate this issue, a modified version of AGF is proposed in this paper. Specifically, the proposed filter first performs noise detection via an image block based noise density estimation and sequential noise density guided rectification on the noise detection result of AGF. Then, a modified Gaussian filter with adaptive variance and window size is used to restore the detected noise pixels. The proposed filter has been extensively evaluated on two representative grayscale images and the Berkeley image dataset BSDS300 with 300 images. Experimental results showed that the proposed filter achieved better denoising effect over the state-of-the-art filters, especially on images with noise-free pixels of gray levels 0 and 255.

Keywords: Gaussian filter, Salt and pepper noise, Image thresholding, Image denoising

1. Introduction

Salt and pepper noise is a common image impulse noise, which is usually generated by bit errors during image acquisition and transmission [1-4]. Salt and pepper noise is one of the main factors causing image contamination and quality deterioration. Each noise pixel has a gray level, either 0 or 255, in an 8-bit image. Noise pixels bring great difficulty for image analysis such as edge detection [5], because noise pixel has similar large gradient value with true edge pixel due to the large differences from its neighboring pixels. Removal of salt and pepper noise is still of great importance to image denoising.

During the past several decades, many filters have been developed to remove salt and pepper noise. These filters can be divided into two categories, i.e., conventional filters [6-8] and switching filters [9-18]. Conventional filters perform image filtering on each pixel, that is, they restore each pixel with a newly estimated gray level. The undesirable filtering on noise-free pixels easily generates blurry image details. Median filter [1] is a representative of conventional filters. It restores each pixel with the median of gray levels in current filtering window. However, when noise density is high, the number of noise pixels may be greater than that of noise-free pixels in a filtering window. In this case, the median value chosen by the median filter is also a noise pixel. Increasing filtering window usually better suppresses severe noise. However, a large filtering window is prone to cause more blurriness. To alleviate this issue, many modified median filters [7-8] have been developed. For example, adaptive median filter (AM) [8] automatically determines filtering window size according to the content of local patch.

Switching filters [9-18] divide image denoising task into two stages, i.e., noise detection and noise restoration. The noise detection stage detects salt and pepper noise pixels. The noise restoration stage restores the detected noise pixels with newly estimated gray levels. Existing switching filters mainly utilize gray level extremes, directionality or their combinations to detect noise pixels. For examples, modified decision based unsymmetric trimmed median filter (UTM) [12], effective restoration method (ERM) [13], adaptive Gaussian filter (AGF) [14] and decision-based coupled window median filter (DBCWMF) [15] directly identify pixels of gray levels 0 and 255 as noise pixels. Differing from UTM, ERM, AGF and DBCWMF, switching median filter with boundary discriminative noise detection (BDND) [16] divides gray levels of a patch into three intervals, and identifies pixels of gray levels in the left or right intervals as noise pixels in two noise detection stages with different patch sizes. Switching-based adaptive weighted mean filter (SAWM) [17] and directional weighted median filter (DWM) [18] respectively define weighted directional gray level differences, and use the minimum difference on different directions to identify noise

pixels. In essence, the minimum directional gray level difference depicts directionality of central pixel in a local patch, i.e., whether the central pixel is located on an edge. Two modified versions (MDWM [19] and MDWF [20]) of DWM employ both gray level extremes and the directionality to identify noise pixels.

For noise restoration, existing switching filters usually use the mean or median variant of filtering window with fixed or adaptive size to restore detected noise pixels. For examples, UTM [12] restores each noise pixel by the mean or median variant of its 3×3 filtering window. BDND [16] restores each noise pixel by the median of remaining neighbors excluding noise in its filtering window with adaptive size. DWM [18] restores each noise pixel by the weighted median of its filtering window. MDWM [19] restores each noise pixel by the weighted median of neighbors without gray levels 0 and 255 on the estimated edge direction of its filtering window. MDWF [20] adaptively restores each noise pixel by the weighted mean of its recursive or non-recursive filtering window. SAWM [17] restores each noise pixel by the weighted mean of adaptive filtering window. ERM [13] restores each noise pixel by the gray level mean of noise-free pixels in its filtering window, where the size of filtering window is adaptively determined according to the ratio of noise pixel number to the total pixel number in the whole image. AGF [14] restores each noise pixel with the weighted mean of 9×9 filtering window, where the weights are defined by a Gaussian filter with adaptive variance. For each noise pixel, DBCWMF [15] first recursively extends its filtering window until noise-free pixels emerge in the filtering window or the filtering window size is equal to 9×9 . Then, if there are noise-free pixels in the filtering window, the median of those noise-free pixels is taken as filtering output, otherwise, the mean of 3×3 filtering window is chosen as the filtering output. In fact, after detecting noise pixels in an image, the subsequent restoration process is similar with image inpainting technique [21-23], because both image denoising and image inpainting usually aim to restore the corrupted pixels by using the uncorrupted neighboring pixels.

Among existing switching filters, AGF usually achieves the best performance in terms of image denoising effect and computational complexity. However, AGF has two main limitations: 1) AGF misclassifies noise-free pixels of gray levels 0 and 255 as noise pixels, thus generating severe false alarm on images with noise-free pixels of gray levels 0 and 255 in noise detection; 2) In the process of noise restoration, AGF uses a Gaussian filter with adaptive variance and fixed filtering window size. The adaptive variance will be affected by severe false alarm in noise detection. In addition, the fixed filtering window is adverse to image restoration. To alleviate both limitations of AGF, we present a modified adaptive Gaussian filter. The proposed filter has three main contributions in comparison with AGF: 1) The proposed filter develops an image block-based noise density estimation method to more accurately estimate noise density of an image; 2) The proposed filter presents a noise detection rectification method to improve the accuracy of noise detection; 3) The proposed filter presents a scheme of adaptively determining the filtering window size to enhance

image restoration effect. Experimental results on two representative grayscale images and Berkeley image dataset BSDS300 with 300 images showed that the proposed filter outperformed the state-of-the-art filters, especially on images with noise-free pixels of gray levels 0 and 255.

The rest of this paper is organized as follows. Section 2 briefly introduces existing adaptive Gaussian filter (AGF) closely related to the proposed filter, and analyzes its advantages and limitations. Section 3 describes theory and implementation of the proposed filter. Experimental results are reported in Section 4. And this paper concludes in Section 5.

2. Existing adaptive Gaussian filter (AGF)

The proposed filter is mainly inspired to remedy the deficiencies of existing adaptive Gaussian filter (AGF) [14] in noise detection and restoration. During exploring existing switching filters, we found that AGF usually has better image denoising performance than other filters. However, AGF still has its limitations. This section will review AGF, and analyze its advantages and limitations.

2.1. Noise detection

For a given pixel $p_{i,j}$ of gray level $f_{i,j}$, AGF directly identifies noise pixels according to $f_{i,j}$. If $f_{i,j}$ is equal to 0 or 255, AGF judges $p_{i,j}$ as a noise pixel, otherwise, AGF judges $p_{i,j}$ as a noise-free pixel. The simple noise detection scheme lets miss detection to be 0. However, it erroneously detects noise-free pixels of gray levels 0 and 255 as noise pixels, thus generating severe false alarm on images with noise-free pixels of gray levels 0 and 255. The detection error lets those noise-free pixels undergo undesirable filtering operation, which may cause image blurriness. In addition, inaccuracy of noise detection caused by false alarm will affect accuracy of noise density estimation and adaptive variance determination in the Gaussian filter for noise restoration.

2.2. Noise restoration

AGF restores each noise pixel with a weighted mean of filtering window with fixed size, where the weights are defined by a Gaussian filter with adaptive variance. The detailed process of noise restoration in AGF is as follows:

- (1) For a detected noise pixel $p_{i,j}$, let Ω denote the $\omega \times \omega$ filtering window centered at $p_{i,j}$, where ω is empirically set to 9.
- (2) Construct a new set U after excluding noise pixels of gray levels 0 and 255 in Ω , and take the weighted mean of pixels in U as the restored gray level, $r_{i,j}$, of $p_{i,j}$, i.e.,

$$r_{i,j} = \frac{\sum_{p_{s,t} \in U} \alpha_{s,t} f_{s,t}}{\sum_{p_{s,t} \in U} \alpha_{s,t}}, \quad (1)$$

where $f_{s,t}$ and $\alpha_{s,t}$ denote the gray level and the weight of $p_{s,t}$. The weight $\alpha_{s,t}$ is defined by a classical Gaussian function as

$$\alpha_{s,t} = \exp\left(-\frac{(s-i)^2 + (t-j)^2}{2\sigma^2}\right), \quad (2)$$

where (i, j) and (s, t) respectively denote the coordinates of $p_{i,j}$ and its neighboring pixel $p_{s,t}$ in the whole image. In addition, the parameter σ is empirically defined as

$$\sigma = \text{noise density} + 0.2, \quad (3)$$

where noise density is estimated to be the ratio between the number of detected noise pixels and the total number of pixels in the whole image.

When restoring a noise pixel, AGF considers space information of neighboring pixels in its filtering window more adequately in comparison with other filters. Studies on natural image statistics [24-26] show that the correlation between two pixels in a local patch decreases with their distance. A pixel is usually more similar to its nearer neighbors than those neighbors far from it. Hence, when restoring a noise pixel, those pixels near it should contribute more than those far from it. Based on this consideration, AGF adopts a Gaussian function with adaptive variance to effectively model unequal weights of noise-free neighbors around being restored noise pixel. As is well known, Gaussian filter is a powerful estimation kernel [27], which is widely used in scale-space representation [28], spatial filtering [29] and Gaussian noise reduction [30].

However, the adaptive variation σ^2 of the Gaussian function defined in Eq. (2) depends on noise density estimation. Unfortunately, accurate noise density estimation will be affected by false alarm on noise-free pixels of gray levels 0 and 255. The detection error on noise-free pixels of gray levels 0 and 255 not only affects calculation of each noise-free neighbor's weight during image restoration, but also loses useful pixel information due to erroneous exclusion of those noise-free neighbors of gray levels 0 and 255. On the other hand, the filtering window with fixed size in AGF is not reasonable for variable noise density. Adaptive variance of the Gaussian function used in AGF can only remedy the deficiency of fixed filtering window size to some extent. However, the size of 9×9 used in AGF may be not large enough to guarantee that there is noise-free pixels in the filtering window when noise density is high. Once U is a null set in Eq. (1), the calculation of restored gray level will be invalid. Conversely, the usage of too large filtering window may cause the loss of localization. Therefore, adaptive filtering window size is a good choice for image restoration.

3. The proposed filter

To remedy the deficiencies of AGF [14] in noise detection and restoration, a modified adaptive Gaussian filter is proposed in this paper. The contributions of the proposed filter are as follows: 1) The proposed filter presents a noise density estimation method, and combines

it with image thresholding technique to rectify noise detection error of AGF on noise-free pixels of gray levels 0 and 255; 2) The proposed filter presents a scheme of adaptively determining the filtering window size to enhance image restoration effect.

3.1. Noise detection

The proposed filter first estimates noise density via an image block based scheme. Then, the estimated noise density guides sequential noise detection. The detailed process of noise density estimation is as follows:

(1) Respectively divide rows and columns of an image into w parts, and accordingly divide an image into $w \times w$ blocks. During this process, if the last part has not the same rows or columns as other parts, we extend the image through mirror-reflecting by enough rows or columns.

(2) Calculate the probability of pixels of gray levels 0 and 255 in each block, and construct a vector $\bar{p} = \{\bar{p}_i, i = 1, \dots, w^2\}$.

(3) Sort all elements in \bar{p} into ascending order and form a vector called $\bar{p}' = \{\bar{p}'_i, i = 1, \dots, w^2\}$.

(4) Estimate noise density of the image according to parity of the image block number w^2 . Specifically, the image noise density is defined as the mean value of middle k elements in \bar{p}' , where k and w^2 have the same parity for the symmetry of taking k elements. The noise density estimation is formulated as

$$D_e = \begin{cases} \frac{1}{k} \sum_{i=-k/2+1}^{k/2} \bar{p}'_{w^2/2+i}, & \text{if } w^2 \text{ is even} \\ \frac{1}{k} \sum_{i=-(k-1)/2}^{(k-1)/2} \bar{p}'_{(w^2+1)/2+i}, & \text{if } w^2 \text{ is odd} \end{cases} \quad (4)$$

The choice of w and k will be discussed in Section 4.1.

After estimating the image noise density, sequential noise detection process is as follows. We use a binary matrix \mathbf{B} to record noise detection result, where 1 and 0 denote noise pixel and noise-free pixel, respectively.

(1) Initialize all elements of \mathbf{B} to zeros, like AGF [14], detect pixels of gray levels 0 and 255 as noise pixels, and set their corresponding elements in \mathbf{B} to ones.

(2) If the estimated noise density D_e defined in Eq. (4) is nearest to one of the first 6 noise densities among $\{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%\}$, continue the following steps, otherwise, the noise detection process stops.

(3) Acquire two image thresholding results

$$seg_0(i, j) = \begin{cases} 1, & \text{if } f_{i,j} = 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$seg_{255}(i, j) = \begin{cases} 1, & \text{if } f_{i,j} = 255 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where 1 and 0 denote object pixel and background pixel, respectively.

(4) Search object pixels in seg_0 and seg_{255} , find out each connected component C_i composed of object pixels under 8-connected graph topology, and construct a set $C = \{C_i, i = 1, \dots, m\}$, where m is the number of connected components.

(5) Remove those components composed of possible noise pixels, and construct a new set

$$C' = \{C_i | n_i > \beta\}, \quad (7)$$

$$\beta = \text{round}(\alpha \times D_e), \quad (8)$$

where n_i is the number of pixels in C_i , and α is a parameter, the choice of which will be discussed in Section 4.1.

(6) Rectify the noise detection result obtained in the first step, i.e., setting B as

$$B(i, j) = 0 \text{ if } B(i, j) = 1 \text{ and } p_{i,j} \in C'. \quad (9)$$

The above noise detection rectification is based on the following considerations: 1) Gray level of a salt and pepper noise pixel can only be 0 or 255; 2) When noise density is not too high, the pixel number of a connected component composed of true noise pixels adhered together should be smaller than that of an image block composed of noise-free pixels of gray levels 0 or 255. The above noise detection rectification is useful for reducing false alarm on images with noise-free pixels of gray levels 0 and 255.

3.2. Noise restoration

The proposed filter first adaptively determines filtering window size according to the content of local patch. Then, the noise restoration scheme of AGF [14] is used to restore detected noise pixels. The detailed process of noise restoration is as follows:

(1) For a given noise pixel $p_{i,j}$, let $w_p \times w_p$ denote the size of local window centered at $p_{i,j}$, initialize $w_p = 3$ and iteratively set $w_p = w_p + 2$ until $N_{uc} \geq 2$ or $w_p \geq W_{\max}$, where N_{uc} denotes the number of detected noise-free (uncorrupted) pixels in current window, and the choice of W_{\max} will be discussed in Section 4.1.

(2) Acquire the $w_p \times w_p$ filtering window Ω centered at $p_{i,j}$, construct a set U after excluding detected noise pixels in Ω . Then, we take the weighted mean of pixels in U as the restored gray level of $p_{i,j}$ according to Eq. (1) defined by AGF. Note that, if U is a null set, we reset $U = \Omega$ to guarantee the validity of Eq. (1). Differing from AGF, the parameter σ in Eq. (2) is redefined as

$$\sigma = D_e + 0.2, \quad (10)$$

where D_e is the noise density estimated by the proposed method in Eq. (4).

4. Experimental results

To evaluate the performance of the proposed filter, we performed extensive experiments on two representative grayscale images and 300 natural images from the Berkeley dataset BSDS300¹. Each noise filter designed for a noisy grayscale image can be easily applied to a noisy color image by applying it to RGB color components separately and merging denoising results of three components as final denoising result of the color image. Therefore, the proposed filter only focuses on noisy grayscale images, and each color image in BSDS300 was converted into grayscale image by using the MATLAB function “rgb2gray”, while keeping the image size unchanged.

The proposed filter is first qualitatively compared with the state-of-the-art salt and pepper noise filters, i.e., AM [8], BDND [16], UTM [12], SAWM [17], MDWF [20], ERM [13], DBCWMF [15] and AGF [14] on two representative grayscale images. Then, image denoising results on the BSDS300 dataset are quantitatively evaluated by two common measures, i.e., peak signal-to-noise ratio (PSNR) [31] and MSSIM (mean structural similarity) [32]:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{N} \sum_{i,j} (r_{i,j} - x_{i,j})^2}, \quad (11)$$

$$\text{MSSIM}(X, Y) = \frac{1}{M} \sum_{j=1}^M \left(\frac{(2\mu_{x_j}\mu_{y_j} + C_1)(2\sigma_{x_j y_j} + C_2)}{(\mu_{x_j}^2 + \mu_{y_j}^2 + C_1)(\sigma_{x_j}^2 + \sigma_{y_j}^2 + C_2)} \right), \quad (12)$$

In Eq. (11), N is the total number of image pixels, $r_{i,j}$ and $x_{i,j}$ denote the gray levels of $p_{i,j}$ in the restored image and the original noise-free image, respectively. In Eq. (12), X and Y are the original noise-free image and the restored image, x_j and y_j are the image contents at the j -th local window, M is the number of local windows, μ_{x_j} and μ_{y_j} are the means of x_j and y_j , σ_{x_j} and σ_{y_j} are their standard deviations, $\sigma_{x_j y_j}$ is the covariance of x_j and y_j , C_1 and C_2 are free parameters. The value of MSSIM is between 0 and 1. Higher PSNR and MSSIM indicate better denoising effect.

In our experiments, the maximum window size of AM was set to 39. The iteration number of MDWF was set to 10. We ran MDWF with different filtering windows of 3×3 , 5×5 , 7×7 , 9×9 , 11×11 , 13×13 , 15×15 on each image, and chosen the result corresponding to the highest PSNR value as the final result. BDND, UTM, SAWM, ERM, DBCWMF and AGF follow their original parameter setting. All testing samples are 8-bit grayscale images. To make each image pixel visit its complete filtering window, we extended four boundaries of an input image through mirror-reflecting to ensure enough pixels.

¹ <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>

4.1. Impact of the parameters

The proposed filter has four parameters, i.e., w , k , α and W_{\max} . The first two parameters (i.e., w and k) are used in our noise density estimation. To evaluate the impact of both parameters on the accuracy of noise density estimation, we specifically recorded average noise density estimation errors obtained by applying the proposed method with different w and k on the BSDS300 dataset corrupted by manually added salt and pepper noise under 9 noise densities varying from 10% to 90% with an incremental step of 10%, where w is selected from 3 to 12. Considering the symmetry of taking k elements in \bar{p}' , when w^2 is odd, k is also odd selected from $\{1, 3, 5, 7, 9, w^2\}$, otherwise, k is even from $\{2, 4, 6, 8, 10, w^2\}$. In fact, taking $k=w^2$ elements indicates calculating the mean of \bar{p}' . When w is odd, taking $k=1$ element indicates calculating the median of \bar{p}' . When w is even, taking $k=2$ elements indicates calculating the median of \bar{p}' . Experimental results under different combinations of w and k are listed in Table 1. The lowest estimation error (0.00130) was obtained with $w=7$ and $k=9$. Hence, w and k are set to 7 and 9 in subsequent experiments, respectively.

Table 1. Average noise density estimation errors under different combinations of w and k

w (odd)	k					
	1	3	5	7	9	w^2
3	0.00167	0.00171	0.00207	0.00256	0.00341	0.00341
5	0.00145	0.00147	0.00148	0.00149	0.00154	0.00338
7	0.00135	0.00133	0.00132	0.00131	0.00130	0.00340
9	0.00135	0.00133	0.00132	0.00132	0.00131	0.00339
11	0.00136	0.00136	0.00136	0.00135	0.00134	0.00335
w (even)	2	4	6	8	10	w^2
4	0.00140	0.00144	0.00151	0.00167	0.00189	0.00339
6	0.00133	0.00133	0.00132	0.00133	0.00134	0.00339
8	0.00135	0.00134	0.00133	0.00132	0.00131	0.00336
10	0.00136	0.00135	0.00134	0.00134	0.00133	0.00334
12	0.00138	0.00137	0.00136	0.00136	0.00135	0.00337

Table 2. Average noise detection error (pixel number) under different α

Noise density	AGF	α								
		100	200	300	400	500	600	700	800	900
10%	799	73	86	94	98	101	106	108	111	114
20%	757	96	106	112	117	121	126	129	130	134
30%	716	117	120	127	131	134	138	142	144	147
40%	673	197	136	140	144	146	148	152	152	154
50%	632	1357	185	162	162	164	164	166	169	169
60%	589	11748	2250	581	269	212	201	198	199	200
70%	547	47948	29127	18591	12260	8207	5582	3949	2763	1912
80%	505	97154	90724	86626	83576	80920	78779	76853	75045	73450

90%	463	130320	129860	129700	129600	129540	129500	129480	129460	129450
The average value of the errors corresponding to the first six noise densities										
	694	2265	480	203	153	146	147	149	151	153

The third parameter α is used in our noise detection stage. It controls sensitivity of distinguishing image blocks composed of noise-free pixels of gray levels 0 and 255 from those blocks composed of noise pixels adhered together. The parameter affects the effectiveness of noise detection rectification on noise-free pixels of gray levels 0 and 255. To evaluate the impact of α on noise detection, we specifically recorded average number of erroneously detected pixels obtained by applying our method with different α on all the images from BSDS300 under 9 noise densities from 10% to 90% with an incremental step of 10%, where α ranges from 100 to 900 with an incremental step of 100. Average noise detection errors (pixel numbers) are listed in [Table 2](#). In addition, to demonstrate the effectiveness of the proposed noise detection rectification, the error of noise detection without rectification operation is also listed in [Table 2](#). In fact, the noise detection without rectification is just the scheme of AGF. From [Table 2](#), one can observe that: 1) when noise density is higher than 60%, the noise detection errors obtained by the proposed filter with all α are higher than the error by AGF. This shows that when noise density is higher than 60%, noise detection rectification should be canceled, because it generates bad effect on noise detection. 2) For average values of the errors corresponding to the first six noise densities, the lowest error (146) was obtained with $\alpha=500$. Hence, α is set to 500, and noise detection rectification is only performed when noise density is nearest to one of the first 6 noise densities among {10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%} in our experiments.

The fourth parameter W_{\max} indicates the maximum permitted filtering window size when iteratively enlarging filtering window for finding noise-free neighboring pixels in the proposed noise restoration stage. To discuss the choice of W_{\max} in detail, we first enlarged the filtering window for each noise pixel in each image until there are more than one noise-free neighbor, and denoted the size of current filtering window as $S_{\max} \times S_{\max}$. Then, we recorded S_{\max} values of all noise pixels in each noisy image, and calculated the number of images whose noise pixels have corresponding S_{\max} values. Experimental results on the BSDS300 dataset corrupted by salt and pepper noise of varying noise density are listed in [Table 3](#). From [Table 3](#), one can observe that when noise density is 90%, there are 300, 22, 15 and 10 noisy images have noise pixels corresponding to $S_{\max}=11, 21, 31$ and 41, respectively.

Table 3. The number of images whose pixels needing different filtering window sizes (S_{\max})

Needed filtering window size (S_{\max})	Noise density								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
11	0	0	0	2	3	6	29	87	300
21	0	0	0	0	0	0	17	17	22
31	0	0	0	0	0	0	14	14	15
41	0	0	0	0	0	0	9	10	10

Table 4. Average PSNR of 29 images obtained by the proposed filter with different W_{\max}

11	21	31	41
18.204	18.936	18.936	18.936

To further evaluate the impact of W_{\max} on noise restoration, we took the 29 noisy images corresponding to $S_{\max}=41$ as testing samples, and set W_{\max} to 11, 21, 31 and 41, respectively. Corresponding experimental results are listed in **Table 4**. **Table 4** shows that average PSNR values obtained by the proposed filter with W_{\max} from $\{21, 31, 41\}$ is higher than that by our filter with $W_{\max}=11$. At the same time, the proposed filter with W_{\max} from $\{21, 31, 41\}$ have the same PSNR values, which is attributed to the usage of the weight defined by Eq. (2) in calculating filtering output of noise pixel. Eq. (2) is a classical Gaussian function, which defines the weight of weighted mean filter defined by Eq. (1). In fact, Eq. (1) can be understood as convolution operation of a Gaussian filtering template H and a local image window U excluding noise pixels, where H can be calculated by Eqs. (1)~(3). Of course, we can also use MATLAB function “fspecial” to generate H , where H under high noise density has more non-zero elements than H under low noise density, because noise density is related with σ in Eq. (3) and the function “fspecial” sets those elements lower than “ $\text{eps} \times \max(H(:))$ ” to zeros due to their very small contributions for filtering output. When we set the size of H to 31×31 and 41×41 , non-zero elements of $H_{31 \times 31}$ and $H_{41 \times 41}$ generated by “fspecial” under the highest noise density 100% only occupy 21 rows and 21 columns at the central location. In other words, $H_{21 \times 21}$, $H_{31 \times 31}$ and $H_{41 \times 41}$ generated by “fspecial” have the same non-zero elements, and these non-zero elements are centered at three templates. In this case, the filtering outputs as the convolution results of respectively using $H_{21 \times 21}$, $H_{31 \times 31}$ and $H_{41 \times 41}$ with U are the same. In summary, too small W_{\max} will affect image denoising effect when noise density is high, which is demonstrated by comparing PSNR values corresponding to $W_{\max}=11$ and $W_{\max}=21$ in **Table 4**. In addition, the proposed filter obtains the same PSNR value for $W_{\max} > 21$ and $W_{\max}=21$, as demonstrated in **Table 4**. Large W_{\max} will add computational time. Therefore, W_{\max} is set to 21 in our experiments, and the choice of W_{\max} is regardless of image size.

4.2. Qualitative comparison

To qualitatively compare image denoising effect of different filters, we chose two widely used 256×256 grayscale images as shown in **Fig. 1** as testing samples, and both images are representatives with and without noise-free pixels of gray levels 0 and 255. The maximum and minimum gray levels of the Lena image are 245 and 23, and there are no noise-free pixels of gray levels 0 and 255. The maximum and minimum gray levels of the rice image are 254 and 0, and there are 3839 noise-free pixels of gray level 0. In our experiments, each image was corrupted by salt and pepper noise with varying noise density from 10% to 90% with an incremental step of 10%. To save space, **Fig. 2** only shows the denoising results under three noise densities, i.e., 10%, 50% and 90%. For the Lena image with noise density

10%, there is no obvious visual difference among the denoising results of different filters. For other two noisy Lena images, AM, BDND and UTM obtain inferior denoising effect than other filters. The results of AM suffer severe blurriness. The results of BDND have a lot of white noise regions. The results of UTM have a number of white and black noise regions. The results of SAWM and MDWF suffer some blurriness, and the results of MDWF and ERM have some white and black noise regions. The results of AGF and DBCWMF under noise density 90% have several white noise regions and black noise regions, respectively. As compared with other filters, the proposed filter obtains better denoising effect, because our results do not have noise regions and preserve more image details.

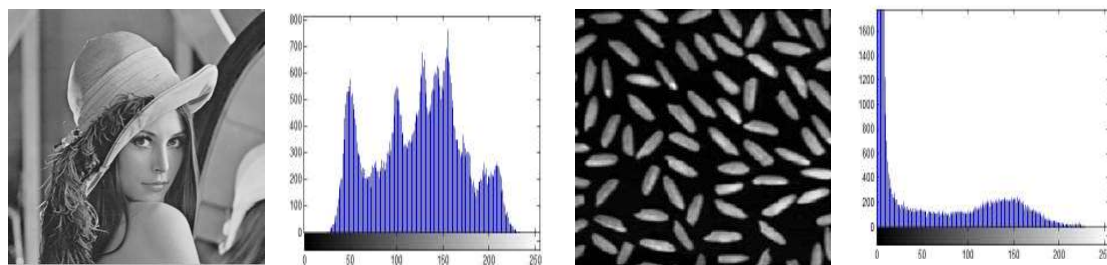


Fig. 1. Two representative grayscale images and their histograms.

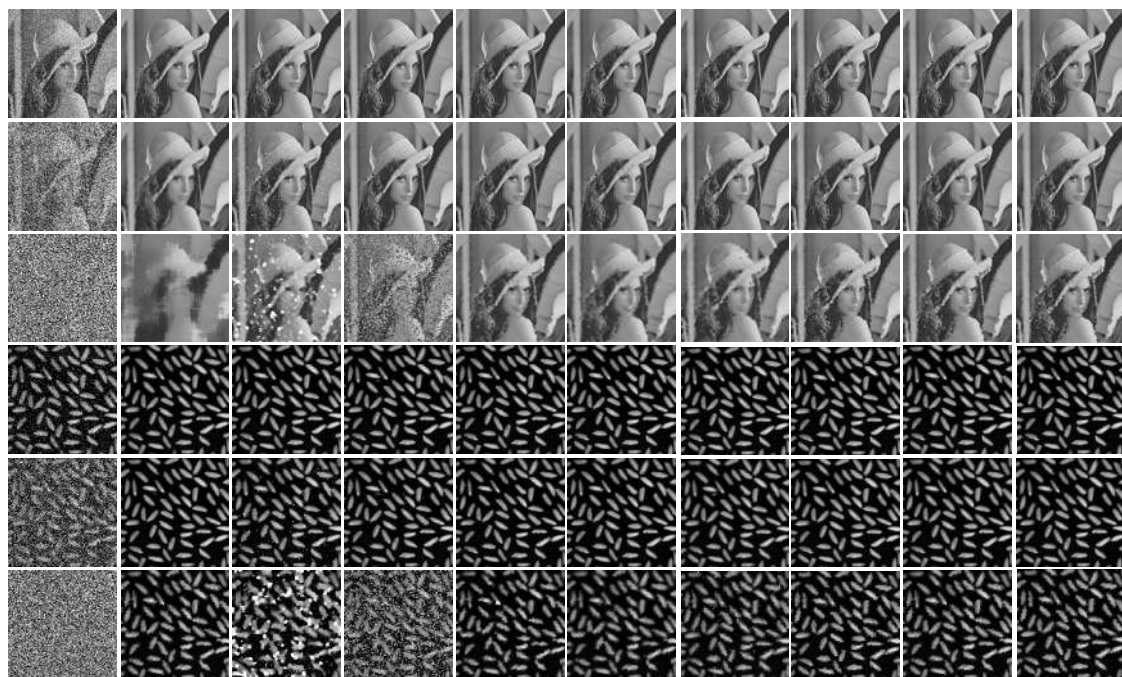


Fig. 2. Visual inspection of denoising results on two representative grayscale images under three noise densities, i.e., 10%, 50% and 90%. Columns 1-8: original images, results of AM [8], results of BDND [16], results of UTM [12], results of SAWM [17], results of MDWF [20], results of ERM [13], results of DBCWMF [15], results of AGF [14], and results of the proposed filter.

For the rice image with noise density 10%, the result of SAWM has a number of black and white noise points. The results of BDND, UTM, ERM and AGF have some white noise regions. The rice boundaries obtained by AM are not smooth enough. MDWF, DBCWMF and the proposed filter obtain the best denoising effect. When noise density is 50%, the results of DBCWMF, AGF and the proposed filter have better visual effect than those of other filters. The results of BDND, UTM and SAWM have a number of white or black noise regions. The results of AM, MDWF and ERM have some white or black noise regions. When noise density is 90%, the results of BDND and UTM have severe noise. The results of AM, MDWF, DBCWMF and AGF have some noise regions. The result of SAWM does not have noise, but its rice boundaries are not smoother than AM, MDWF, DBCWMF, AGF and the proposed filter. The proposed filter achieves better denoising effect on the rice images under noise densities 50% and 90% in comparison with other filters, because our results do not have noise, and rice boundaries are more complete.

4.3. Quantitative comparison

To provide quantitative comparison of image denoising effect of different filters, we evaluated them on the BSDS300 dataset using two measures, i.e., PSNR and MSSIM. Each testing image was also randomly corrupted by salt and pepper noise with varying density from 10% to 90% with an incremental step of 10%. To more adequately compare denoising effects of different filters on images with and without noise-free pixels of gray levels 0 and 255, we specifically calculated average PSNR and MSSIM values on two types of images under each noise density, where the numbers of images belonging to both types are 138 and 162, respectively. Furthermore, we also recorded average PSNR and MSSIM values on the whole dataset.

Fig. 3 shows the box-whisker plots of PSNR and MSSIM. Furthermore, **Fig. 4** shows the bar charts of average PSNR and MSSIM values obtained by different filters on three groups of images, i.e., images without noise-free pixels of gray levels 0 and 255, images with noise-free pixels of gray levels 0 and 255, and the whole BSDS300 dataset. From **Figs. 4 (a)-(d)**, it can be observed that the proposed filter and AGF obtain higher PSNR and MSSIM values than other filters on two types of images. For images without noise-free pixels of gray levels 0 and 255, the proposed filter obtains slightly higher PSNR and MSSIM values than AGF. For images with noise-free pixels of gray levels 0 and 255, the proposed filter achieves significant improvement in terms of PSNR and MSSIM in comparison with other filters, showing large superiority on this type of images. **Figs. 4 (e)-(f)** show that the proposed filter obtains the highest average PSNR and MSSIM values on the whole BSDS300 dataset, thus demonstrating better denoising effect than its counterparts.

4.4 Computational time

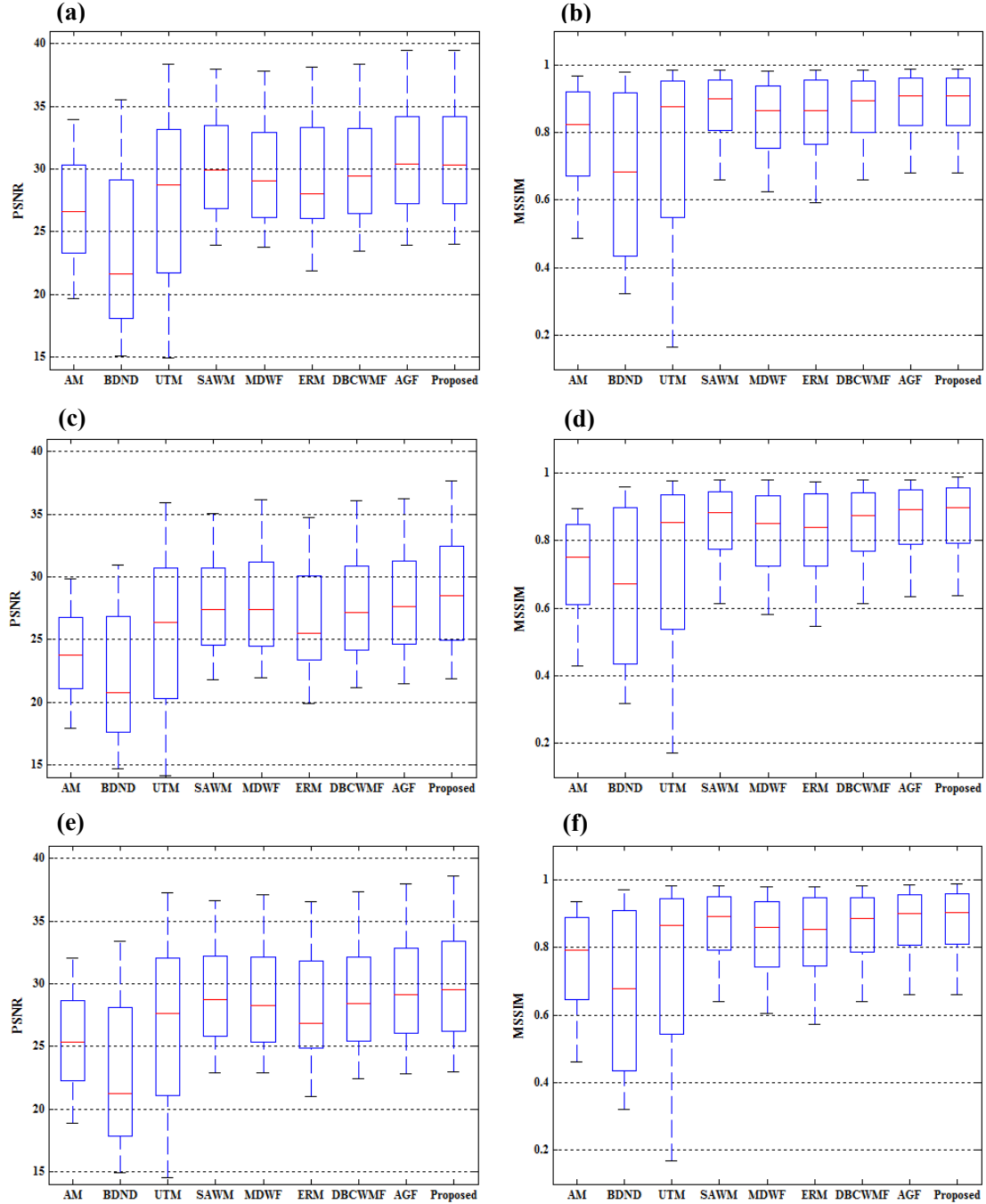


Fig. 3. Box-whisker plots of average PSNR and MSSIM values obtained by applying AM [8], BDND [16], UTM [12], SAWM [17], MDWF [20], ERM [13], DBCWMF [15], AGF [14] and the proposed filter to three groups of images, i.e., (a)-(b): 162 images without noise-free pixels of gray levels 0 and 255, (c)-(d) 138 images with noise-free pixels of gray levels 0 and 255, and (e)-(f) 300 images from the BSDS300 dataset under 9 noise densities.

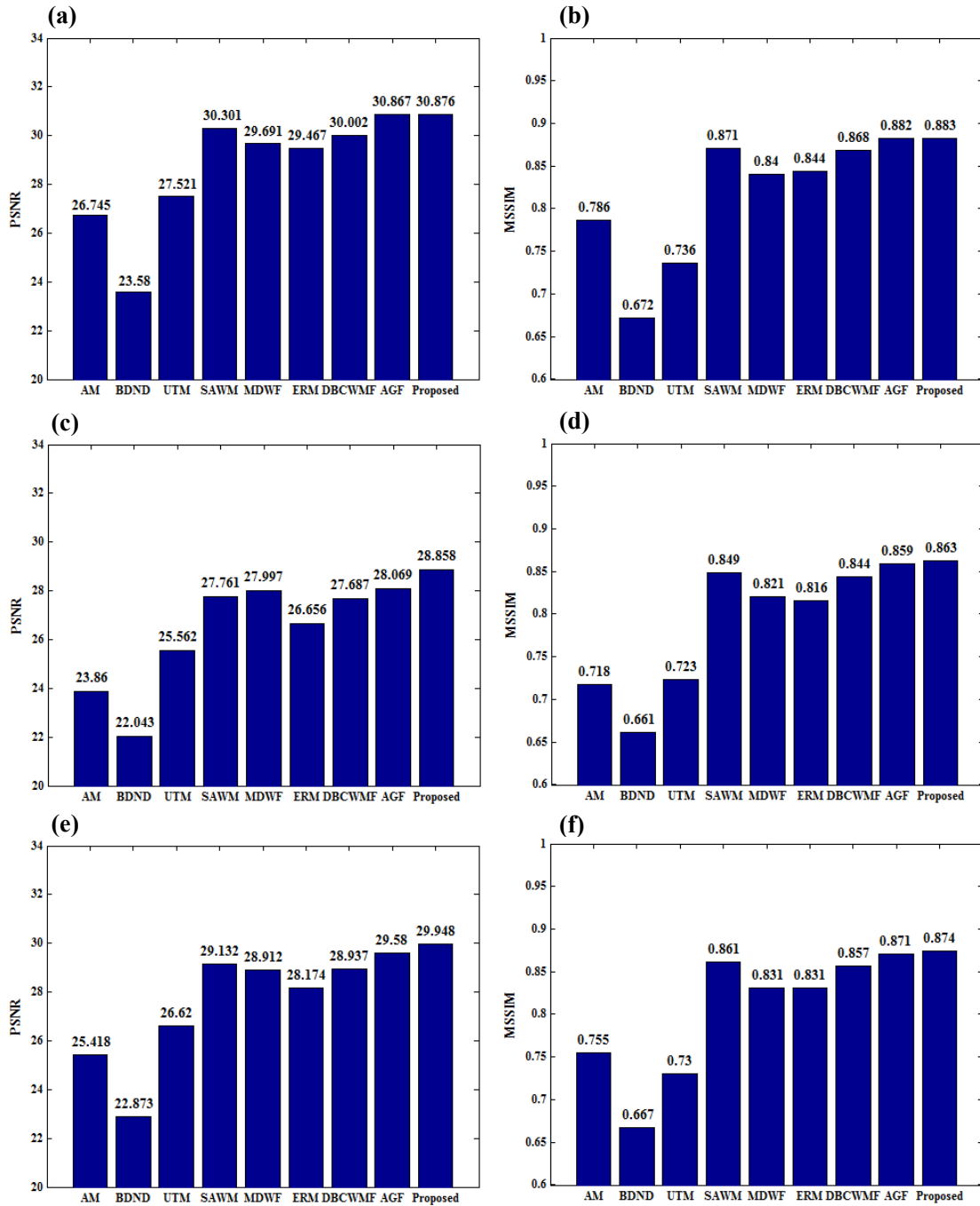


Fig. 4. Bar chart of the PSNR and MSSIM mean values obtained by applying AM [8], BDND [16], UTM [12], SAWM [17], MDWF [20], ERM [13], DBCWMF [15], AGF [14] and the proposed filter to three groups of images, i.e., (a)-(b): 162 images without noise-free pixels of gray levels 0 and 255, (c)-(d): 138 images with noise-free pixels of gray levels 0 and 255, and (e)-(f): 300 images from the BSDS300 dataset under 9 noise densities.

Table 5. Average computational time (in seconds) on BSDS300 under 9 noise densities

AM	BDND	UTM	SAWM	MDWF	ERM	DBCWME	AGF	Proposed
9.97	13.08	4.02	30.06	81.10	3.87	4.58	1.44	26.90

The computational time of the proposed filter is related with several factors such as image size and noise density. For example, the average total computational time is around 5s for the denoising of each noisy image with noise density 10% from the BSDS300 dataset on our Linux server with 1 CPU and 2G memory. To compare the computational efficiency of various filters, we specifically recorded their average computational time on BSDS300 under 9 noise densities. Experimental results are listed in **Table 5**. **Table 5** shows that the proposed filter is faster than SAWM and MDWF, but is slower than other counterparts. Overall, the proposed filter achieves satisfactory image denoising results within reasonable computational time.

5. Conclusions

In this paper, we have proposed a modified adaptive Gaussian filter for the removal of salt and pepper noise. As compared with existing adaptive Gaussian filter (AGF), the proposed filter presents an image block based scheme to accurately estimate noise density of an image, and combines noise density estimation result and image thresholding results obtained by gray levels 0 and 255 to rectify the noise detection error of AGF. In addition, when restoring detected noise pixels, the proposed filter adopts adaptive filtering window size instead of empirically fixed window size in AGF. Experimental results on two representative grayscale images and the BSDS300 dataset showed that the proposed filter dramatically improved denoising effect in comparison with the state-of-the-art filters, especially on images with noise-free pixels of gray levels 0 and 255.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Nos. 61202318, 61202272, 61202313, 61203244 and 61370163), Fujian College Outstanding Young Research Personnel Training Project (No. JA13247), Technology Project of Provincial University of Fujian Province (No. JK2014040), Nature Science Foundation of Jiangsu Province (No. BK20131342), Guangxi Natural Science Fund of China (No. 2011GXNSFB018070) and Scientific Research Project of provincial University of Shandong Province (No. J15LN08).

References

- [1] A. Bovik, Handbook of Image and Video Processing, New York: Academic, 2000.

[Article \(CrossRef Link\)](#)

- [2] H. H. Chou and L. Y. Hsu, "A noise-ranking switching filter for images with general fixed-value impulse noises," *Signal Processing*, vol. 106, pp. 198–208, January, 2015.
[Article \(CrossRef Link\)](#)
- [3] Y. Liu, C. Zuo and X. Tan, "A kalman filter based video denoising method using intensity and structure tensor," *KSI Transactions on Internet and Information Systems*, vol. 8, no. 8, pp. 2866–2880, August, 2014. [Article \(CrossRef Link\)](#)
- [4] F. Ahmed and S. Das, "Removal of high-density salt-and-pepper noise in images with an iterative adaptive fuzzy filter using alpha-trimmed mean," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 5, pp. 1352–1358, October, 2014. [Article \(CrossRef Link\)](#)
- [5] P. Y. Chen and C. Y. Lien, "An efficient edge-preserving algorithm for removal of salt-and-pepper noise," *IEEE Signal Processing Letters*, vol. 15, pp. 833–836, December, 2008.
[Article \(CrossRef Link\)](#)
- [6] S. C. Hsia and C. L. Chen, "A fast efficient restoration algorithm for high-noise image filtering with adaptive approach," *Journal of Visual Communication and Image Representation*, vol. 16, no. 3, pp. 379–392, June, 2005. [Article \(CrossRef Link\)](#)
- [7] T. S. Huang, G. J. Yang and G. Y. Tang, "Fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 1, no. 1, pp. 13–18, February, 1979.
[Article \(CrossRef Link\)](#)
- [8] R. H. Chan, C. W. Ho and M. Nikolova, "Salt and pepper noise removal by median type noise detectors and detail-preserving regularization," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, October, 2005. [Article \(CrossRef Link\)](#)
- [9] C. H. Lin, J. S. Tsai and C. T. Chiu, "Switching bilateral filter with a texture/noise detector for universal noise removal," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2307–2320, September, 2010. [Article \(CrossRef Link\)](#)
- [10] K. K. V. Toh and N. A. M. Isa, "Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 281–284, March, 2010.
[Article \(CrossRef Link\)](#)
- [11] H. X. Xu, G. X. Zhu and H. Y. Peng, "Adaptive fuzzy switching filter for images corrupted by impulse noise," *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1657–1663, November, 2004.
[Article \(CrossRef Link\)](#)
- [12] S. Esakirajan, T. Veerakumar, A. N. Subramanyam and C. H. PremChand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 287–290, May, 2011.
[Article \(CrossRef Link\)](#)
- [13] Z. M. Ramadan, "Efficient restoration method for images corrupted with impulse noise," *Circuits System and Signal Processing*, vol. 31, no. 4, pp. 1397–1406, August, 2012.
[Article \(CrossRef Link\)](#)

- [14] M. Nasri, S. Saryazdi and H. Nezamabadi-pour, "A fast adaptive salt and pepper noise reduction method in images," *Circuits System and Signal Processing*, vol. 32, no. 4, pp. 1839–1857, August, 2013. [Article \(CrossRef Link\)](#)
- [15] V. S. Bhadouria, D. Ghoshal, A. H. Siddiqi, "A new approach for high density saturated impulse noise removal using decision-based coupled window median filter," *Signal, Image and Video Processing*, vol. 8, no. 1, pp. 71–84, January, 2014. [Article \(CrossRef Link\)](#)
- [16] P. E. Ng and K. K. Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1506–1516, June, 2006. [Article \(CrossRef Link\)](#)
- [17] X. Zhang and Y. Xiong, "Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter," *IEEE Signal Processing Letters*, vol. 16, no. 4, pp. 295–298, April, 2009. [Article \(CrossRef Link\)](#)
- [18] Y. Dong and S. Xu, "A new directional weighted median filter for removal of random-valued impulse noise," *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 193–196, March, 2007. [Article \(CrossRef Link\)](#)
- [19] C. T. Lu and T. C. Chou, "Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter," *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1287–1295, July, 2012. [Article \(CrossRef Link\)](#)
- [20] Z. Li, G. Liu, Y. Xu and Y. Cheng, "Modified directional weighted filter for removal of salt & pepper noise," *Pattern Recognition Letters*, vol. 40, no. 1, pp. 113–120, April, 2014. [Article \(CrossRef Link\)](#)
- [21] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image inpainting," in *Proc. of the ACM SIGGRAPH Conf.on Computer Graphics*, July 23–28, pp. 417–424, 2000. [Article \(CrossRef Link\)](#)
- [22] C. Qin, S. Wang, X. P. Zhang, "Simultaneous inpainting for image structure and texture using anisotropic heat transfer model," *Multimedia Tools and Applications*, vol. 56, no. 3, pp. 469–483, February, 2012. [Article \(CrossRef Link\)](#)
- [23] C. Qin, F. Cao and X. P. Zhang, "Efficient image inpainting using adaptive edge-preserving propagation," *The Imaging Science Journal*, vol. 59, no. 4, pp. 211–218, August, 2011. [Article \(CrossRef Link\)](#)
- [24] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 1193–1216, March, 2001. [Article \(CrossRef Link\)](#)
- [25] A. Torralba and A. Oliva, "Statistics of natural image categories," *Network: Computation in Neural System*, vol. 14, no. 3, pp. 391–412, May, 2003. [Article \(CrossRef Link\)](#)
- [26] A. Hyvärinen, J. Hurri and P.O. Hoyer, *Natural image statistics: a probabilistic approach to early computational vision*, Springer, New York, 2009. [Article \(CrossRef Link\)](#)
- [27] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern classification*, Wiley, New York, 2001. [Article \(CrossRef Link\)](#)
- [28] T. Lindeberg, *Scale-space theory in computer vision*, Springer, Berlin, 1994.

[Article \(CrossRef Link\)](#)

- [29] R. C. Gonzalez, R. E. Woods, *Digital image processing (2nd edition)*, Prentice Hall, New York, 2002. [Article \(CrossRef Link\)](#)
- [30] J. Portilla, V. Strela, M.J. Wainwright and E.P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, November, 2003. [Article \(CrossRef Link\)](#)
- [31] N. I. Petrović and V. Crnojević, "Universal impulse noise filter based on genetic programming," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1109–1120, July, 2008. [Article \(CrossRef Link\)](#)
- [32] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April, 2004. [Article \(CrossRef Link\)](#)



Zuoyong Li received the B.S. and M.S. degrees of Computer Science and Technology from Fuzhou University, Fuzhou, PR China, in 2002 and 2006. He received the Ph.D. degree from the School of Computer Science and Technology at Nanjing University of Science and Technology, Nanjing, PR China, in 2010. He is currently an associate professor in Department of Computer Science of Minjiang University, Fuzhou, China. He has published more than 10 papers in international/national journals. His current research interest is image processing.



Kezong Tang received his M.S. degree in pattern recognition and intelligent systems from Jiangsu University of Science and Technology in 2007. He received his Ph.D. degree from the School of Computer Science and Technology at Nanjing University of Science and Technology, Nanjing, PR China, in 2012. Currently, he is an associate professor in School of Information Engineering, Jingdezhen Ceramic Institute. His research interests include image processing and artificial intelligence.



Yong Cheng received his M.S. degree in computer science and technology from Nanjing University of Information Science & Technology, China, in 2004. He received a Ph.D. degree in computer applications from Nanjing University of Science & Technology, China, in 2011. He worked as a postdoctoral researcher at Southeast University, China. His research interests are in image processing and pattern recognition.



Xiaobo Chen received the BS degree in computer science and MS degree in pattern recognition and intelligence systems both from Jiangsu University in 2004 and 2007, respectively. He earned PhD degree in pattern recognition and intelligence systems from Nanjing University of Science and Technology (NUST). From March to August, 2011, he served as a research assistant at Department of Computing, the Hong Kong Polytechnic University. From April 2015, he served as PostDoc research associate at School of Medicine, the University of North Carolina at Chapel Hill, NC, USA. He is also a lecturer at Jiangsu University. He has published more than 20 technical papers in the area of pattern recognition and machine learning which have been cited more than 100 times. He also serves as a reviewer for more than 10 international journals, including IEEE Transactions on Neural Networks and Learning Systems, Pattern Recognition, Neural Networks etc. He received the best student paper award in the 4th Chinese Conference on Pattern Recognition and the excellent doctor degree dissertation award from NUST.



Chongbo Zhou received his B.S. degree from Qufu Normal University, China, in 2002, M.S. degree from Shanghai University, China, in 2005, and Ph.D. degree in Computer Science from Nanjing University of Science and Technology, China, in 2013. He is currently a lecturer in Information Processing Research Group of Qufu Normal University. His research interests include image segmentation and pattern recognition.