



## Low-density noise removal based on lambda multi-diagonal matrix filter for binary image

Li Li<sup>1,2</sup> · Hongwei Ge<sup>1,2</sup> · Yixin Zhang<sup>2</sup> · Jianqiang Gao<sup>3</sup>

Received: 10 May 2016/Accepted: 8 August 2016/Published online: 19 August 2016  
© The Natural Computing Applications Forum 2016

**Abstract** Binary image denoising is a well-known problem, and it is the concern of diverse application areas. Many classical denoising techniques have evolved over the years, such as mean filter, median filter, and morphological filter. A new denoising method based on the multiplication of lambda multi-diagonal binary matrix ( $\lambda$ -MDBM) is proposed for binary images in this paper. In proposed method, first the users need to choose an appropriate lambda value from the set  $\{\lambda | \lambda \in [0.5, 1]\}$ , and then the hybrid noisy binary image matrix can be obtained by adding the noisy binary image matrix times the  $\lambda$ -MDBM and the transpose of the noisy binary image matrix times the  $\lambda$ -MDBM, and finally the de-noised binary image can be obtained by a preset threshold. The experimental results show that a new denoising method based on  $\lambda$ -MDBM is possible in peak signal-to-noise ratio and mean square error.

**Keywords** Image denoising · Filtering · Binary image processing · Lambda multi-diagonal matrix filter

### 1 Introduction

Digital image processing method is widely used in all kinds of fields, such as face detection and recognition [1–12], tracking [13–16], and biomedical image processing [17–19]. Binary image denoising techniques are very important in modern digital image processing. Many denoising algorithms had been proposed, such as information entropy filter [20], optimization of partition-based weighted sum filters [21], nonlocal transform-domain filter [22], based on wavelet filters [23–26], and based on independent component analysis and hierarchical fusion hybrid filter [27]. The performance of binary image processing can be affected by the presence of salt and pepper noise [28, 29].

Min and Max filters are two classical denoising methods, which can remove salt and pepper noise, respectively. However, the min and max filters can not be work when both types of noise are included in the binary image. Therefore, many researchers proposed all kinds of median filters and morphological operations-based filters. Median filter is a kind of nonlinear filters, and it has good abilities of keeping edge. But, it requires sorting within the pixel neighbourhood specified by the filter size. The computational burden will increase as the size of the filter increases. According to the structuring element, morphological operations erosion and dilation can be applied to remove either salt or pepper noise. In addition, the combination of erosion and dilation operations can be used to remove noise with salt and pepper both types. In [30], Liu et al. used Laplacian regularization denoising. In [31], Anbarjafari et al. proposed a multi-diagonal matrix filter for binary image denoising. But, the authors only considered a

✉ Jianqiang Gao  
dr.jq.gao@gmail.com; jianqianggao@126.com

Li Li  
liliiot@163.com

Hongwei Ge  
ghw8601@163.com

Yixin Zhang  
zyx@jiangnan.edu.cn

<sup>1</sup> Key Laboratory of Advanced Process Control for Light Industry, Jiangnan University, Ministry of Education, Wuxi 214122, China

<sup>2</sup> School of Internet of Things, Jiangnan University, Wuxi 214122, Jiangsu, China

<sup>3</sup> College of Computer and Information, Hohai University, Nanjing 210098, China

special case. The pixel in the hybrid noisy binary image is replaced with a new value which is the sum of its horizontal and vertical neighbourhood. However, it is not very accurate in terms of Anbarjafari et al. proposed elements of main diagonal is 1 in the multi-diagonal matrix. Therefore, in this paper, an improved method was proposed. The proposed approach is a three-step technique: first, the users need to choose an appropriate lambda value from the set  $\{\lambda | \lambda \in [0.5, 1]\}$ , and then the hybrid noisy binary image matrix can be obtained by adding the noisy binary image matrix times the  $\lambda$ -MDBM and the transpose of the noisy binary image matrix times the  $\lambda$ -MDBM, and finally the hybrid noisy binary image denoising can be achieved by processing preset threshold.

The remainder of this paper is organized as follows. Section 2 describes the proposed approach in detail. Section 3 reports the experimental results. Section 4 concludes this paper.

## 2 Lambda multi-diagonal matrix filter

### 2.1 Concept of lambda multi-diagonal matrix

A lambda  $\xi$ -diagonal matrix is defined to be a symmetric square matrix,  $\psi_{k \times k}(\lambda, \xi)$ , which has the value  $\lambda$  on its main diagonal, the value 1 on its  $\xi$ -diagonals, and 0 on the remaining members [31]. The mathematical formulation can be written as Eq. (1).

$$\psi_{k \times k}(\lambda, \xi) = \begin{cases} \lambda, & i = j \\ 1, & |i - j| \leq \xi \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where  $i, j = 1, \dots, k$ ,  $3 \leq \xi < k$ ,  $\xi$  is odd,  $\psi_{k \times k}(\lambda, \xi)$  denotes a lambda  $\xi$ -diagonal matrix,  $k$  denotes the size of square matrix  $\psi_{k \times k}(\lambda, \xi)$ ,  $\lambda$  denotes input value by users from the set  $\{\lambda | \lambda \in [0.5, 1]\}$ , and  $\xi$  denotes the total number of diagonals including the main diagonal,  $(\xi - 1)/2$  above and  $(\xi - 1)/2$  below of the main diagonal.

Generally speaking, a lambda  $\xi$ -diagonal matrix can be denoted as follows:

$$\psi_{k \times k}(\lambda, \xi) = \begin{pmatrix} \lambda & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & & & & & & & & & \\ 0 & \cdots & 0 & 0 & \cdots & 1 & \cdots & \lambda & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & & & & & & & & & \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 1 & \cdots & \lambda \end{pmatrix} \quad (2)$$

where the number of  $\lambda \cdots 1$  is  $(\xi + 1)/2$ , the number of  $1 \cdots \lambda \cdots 1$  is  $\xi$ , and the number of  $1 \cdots \lambda$  is  $(\xi + 1)/2$ . In order to describe the lambda  $\xi$ -diagonal matrix more clearly, we give some examples as below.

**Case 1**  $k = 7, \xi = 3$

$$\psi_{7 \times 7}(\lambda, 3) = \begin{pmatrix} \lambda & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & \lambda & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & \lambda & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \lambda & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \lambda & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & \lambda & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & \lambda \end{pmatrix} \quad (3)$$

**Case 2**  $k = 7, \xi = 5$

$$\psi_{7 \times 7}(\lambda, 5) = \begin{pmatrix} \lambda & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \lambda & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & \lambda & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & \lambda & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & \lambda & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & \lambda & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & \lambda \end{pmatrix} \quad (4)$$

**Case 3**  $k = 8, \xi = 3$

$$\psi_{8 \times 8}(\lambda, 3) = \begin{pmatrix} \lambda & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \lambda & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \lambda & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \lambda & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \lambda & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \lambda & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \lambda & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \lambda \end{pmatrix} \quad (5)$$

**Case 4**  $k = 8, \xi = 5$

$$\psi_{8 \times 8}(\lambda, 5) = \begin{pmatrix} \lambda & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & \lambda & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & \lambda & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & \lambda & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & \lambda & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & \lambda & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \lambda & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \lambda \end{pmatrix} \quad (6)$$

**Case 5**  $k = 8, \xi = 7$

$$\psi_{8 \times 8}(\lambda, 7) = \begin{pmatrix} \lambda & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \lambda & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & \lambda & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & \lambda & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & \lambda & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & \lambda & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & \lambda & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \lambda \end{pmatrix} \quad (7)$$

## 2.2 Lambda $\xi$ -diagonal matrix multiplication based filtering

Let us consider a binary image matrix  $A_{m \times n} \in R^{m \times n}$ , and  $\psi_{n \times n}(\lambda, \xi)$  as follows:

$$A_{m \times n} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,j-1} & a_{1,j} & a_{1,j+1} & \cdots & a_{1,n} \\ \vdots & & & & & & \\ a_{i-1,1} & \cdots & a_{i-1,j-1} & a_{i-1,j} & a_{i-1,j+1} & \cdots & a_{i-1,n} \\ a_{i,1} & \cdots & a_{i,j-1} & a_{i,j} & a_{i,j+1} & \cdots & a_{i,n} \\ a_{i+1,1} & \cdots & a_{i+1,j-1} & a_{i+1,j} & a_{i+1,j+1} & \cdots & a_{i+1,n} \\ \vdots & & & & & & \\ a_{m,1} & \cdots & a_{m,j-1} & a_{m,j} & a_{m,j+1} & \cdots & a_{m,n} \end{pmatrix} \quad (8)$$

$$\Omega_1 = A_{m \times n} \psi_{n \times n}(\lambda, 3) = \begin{pmatrix} \lambda a_{1,1} + a_{1,2} & \cdots & a_{1,j-1} + \lambda a_{1,j} + a_{1,j+1} & \cdots & a_{1,n-1} + \lambda a_{1,n} \\ \vdots & & & & \\ \lambda a_{i,1} + a_{i,2} & \cdots & a_{i,j-1} + \lambda a_{i,j} + a_{i,j+1} & \cdots & a_{i,n-1} + \lambda a_{i,n} \\ \vdots & & & & \\ \lambda a_{m,1} + a_{m,2} & \cdots & a_{m,j-1} + \lambda a_{m,j} + a_{m,j+1} & \cdots & a_{m,n-1} + \lambda a_{m,n} \end{pmatrix} \quad (10)$$

In  $\Omega_1$ , each element of  $A$  has been replaced by addition of the pixel with its left and right neighbours in a row. Similarly, the result of multiplication of  $A^T$  by  $\psi_{m \times m}(\lambda, 3)$  can be obtained. Where  $A^T$  denotes the transpose of  $A$ .

$$\Omega_2 = A_{m \times n}^T \psi_{m \times m}(\lambda, 3) = \begin{pmatrix} \lambda a_{1,1} + a_{2,1} & \cdots & a_{j-1,1} + \lambda a_{j,1} + a_{j+1,1} & \cdots & a_{n-1,1} + \lambda a_{n,1} \\ \vdots & & & & \\ \lambda a_{1,i} + a_{2,i} & \cdots & a_{j-1,i} + \lambda a_{j,i} + a_{j+1,i} & \cdots & a_{n-1,i} + \lambda a_{n,i} \\ \vdots & & & & \\ \lambda a_{1,m} + a_{2,n} & \cdots & a_{j-1,m} + \lambda a_{j,m} + a_{j+1,m} & \cdots & a_{n-1,m} + \lambda a_{n,m} \end{pmatrix} \quad (11)$$

Each element of  $\Omega_2$  is addition of a pixel with its neighbours in a row of  $A^T$ . The new matrix  $\Omega$  can be obtained by adding  $\Omega_1$  and  $\Omega_2^T$  as follows:

$$\Omega = \Omega_1 + \Omega_2^T = \begin{pmatrix} 2\lambda a_{1,1} + a_{1,2} + a_{2,1} & \cdots & a_{1,j-1} + 2\lambda a_{1,j} + a_{2,j} + a_{1,j+1} & \cdots & a_{1,n-1} + a_{2,n} + 2\lambda a_{1,n} \\ \vdots & & & & \\ 2\lambda a_{i,1} + a_{i-1,1} + a_{i,2} + a_{i+1,1} & \cdots & a_{i,j-1} + a_{i-1,j} + 2\lambda a_{i,j} + a_{i+1,j} + a_{i,j+1} & \cdots & a_{i,n-1} + 2\lambda a_{i,n} + a_{i-1,n} + a_{i+1,n} \\ \vdots & & & & \\ 2\lambda a_{m,1} + a_{m-1,1} + a_{m,2} & \cdots & a_{m,j-1} + 2\lambda a_{m,j} + a_{m-1,j} + a_{m,j+1} & \cdots & a_{m,n-1} + 2\lambda a_{m,n} + a_{m-1,n} \end{pmatrix} \quad (12)$$

$$\psi_{n \times n}(\lambda, \xi) = \begin{pmatrix} \lambda & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & & & & & & & & \\ 0 & \cdots & 0 & 0 & \cdots & 1 & \cdots & \lambda & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & & & & & & & & \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 1 & \cdots & \lambda \end{pmatrix} \quad (9)$$

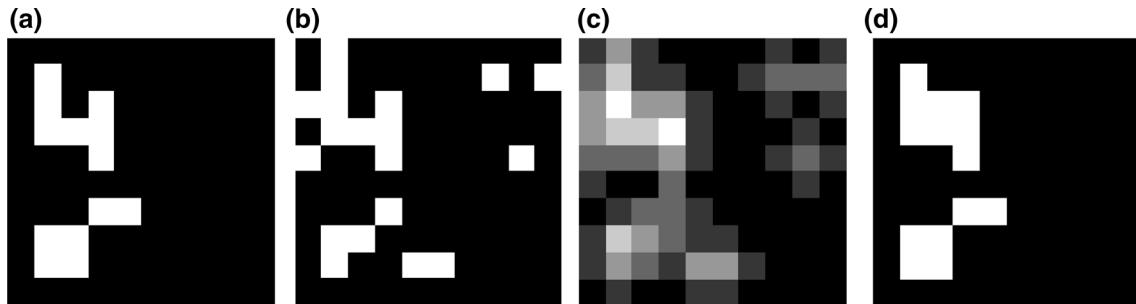
To show the proposed approach, we set  $\xi = 3$ . The result of multiplication of  $A_{m \times n}$  by  $\psi_{n \times n}(\lambda, 3)$  is shown as below:

From an image point of view, we can regard the matrix as a binary image in terms of Eq. (12). And if  $a_{i,j}$  denotes the pixels of this image, it is not hard to find that the pixel is replaced with a new value which is the sum of its horizontal and vertical neighbourhood. In addition, from the viewpoint of mathematics, the number of pixels in each neighbourhood direction (from above to below and from left to right) is a function of  $\lambda$  and  $\xi$  (viz.  $f(\lambda, \xi)$ ). The optimization problem of  $f(\lambda, \xi)$  needs to be solved when we want to obtain the original pixels from the noise image. However, it is complicated.

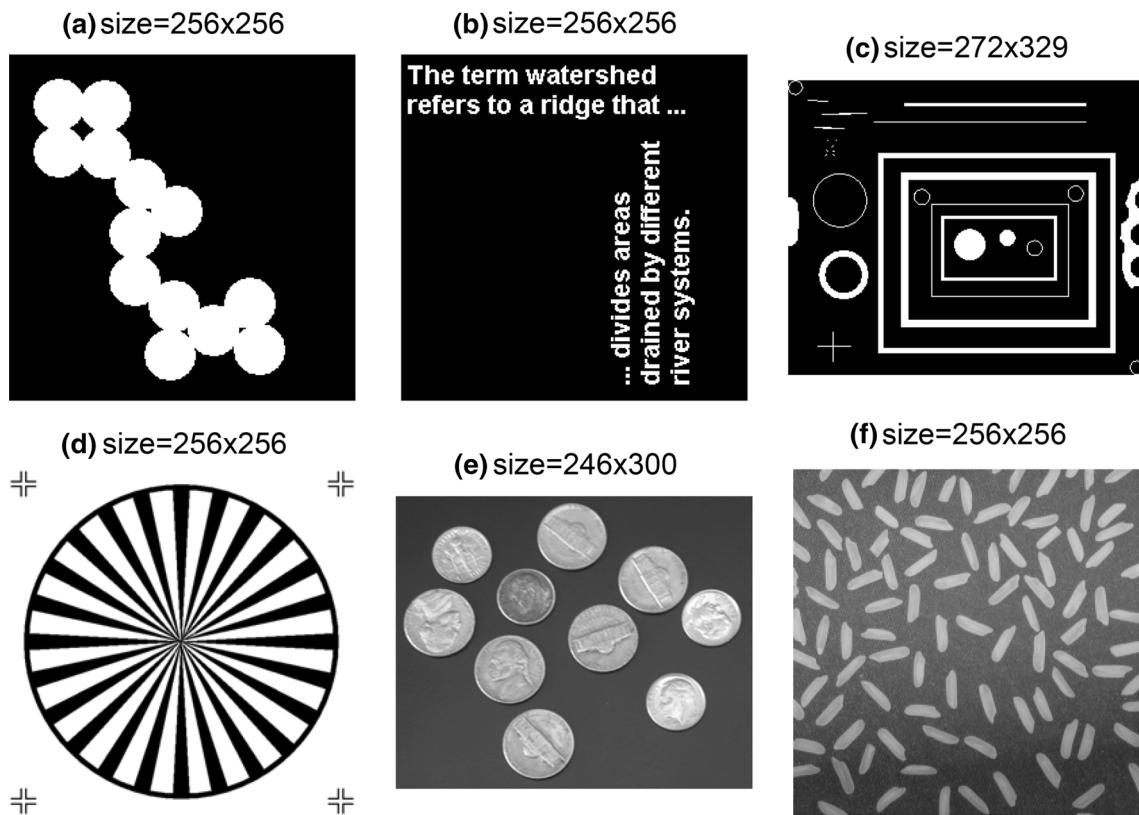
According to [31], each pixel of a connected component with 4-connectivity in  $\Omega$  will accumulate a value less than  $\xi$ , if and only if the width and height of the component is less than  $\xi$ . Generally speaking, the threshold value can be set  $\xi$ . The connected components with sizes less than  $\xi$  will be removed. The mathematical formulation will show how each pixel will be thresholded to generate the final image B, as Eq. (13).

$$B = \begin{cases} 1 & Q(i,j) \geq \xi \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where  $1 \leq i \leq m, 1 \leq j \leq n$ . When the  $\lambda$  and  $\xi$  are available, the final de-noised image B can be obtained according to Eq. (13). In [31], there is an important parameter  $\lambda$  has been neglected by Anbarjafari. An example is given to explain this point. If the pixel  $a_{ij}$  is 0, for  $\lambda \in [0.5, 1]$ , we have  $a_{ij-1} + a_{i-1,j} + 2a_{ij} + a_{i+1,j} + a_{ij+1} = a_{ij-1} + a_{i-1,j} + 2\lambda a_{ij} + a_{i+1,j} + a_{ij+1}$  ([31] proposed method). On the contrary, if the pixel  $a_{ij}$  is 1, we have  $a_{ij-1} + a_{i-1,j} + 2a_{ij} + a_{i+1,j} + a_{ij+1} \geq a_{ij-1} + a_{i-1,j} + 2\lambda a_{ij} + a_{i+1,j} + a_{ij+1}$ . Hence, the parameter  $\lambda$  is very important in Eq. (13). Therefore, the calculation flow of proposed algorithm as shown below.



**Fig. 1** **a** Original binary image, **b** corrupted image, **c** the generated intermediate image using Eq. (12) and **d** filter image using Eq. (13) with  $\lambda = 1, \xi = 3$



**Fig. 2** Original images for the evaluation test: **a** circles, **b** text, **c** blobs, **d** testpat1, **e** coins, and **f** rice

**Algorithm 1** Proposed algorithm

---

**Input:** Binary image with noise  $A$  and appropriate  $\lambda$  value from the set  $\{\lambda | \lambda \in [0.5, 1]\}$ .  
**Step 1.** The  $\Omega_1$  can be obtained according to eq.(10) with  $\xi$ ;  
**Step 2.** The  $\Omega_2$  can be obtained according to eq.(11) with  $\xi$ ;  
**Step 3.** The  $\Omega$  can be obtained according to eq.(12);  
**Output:** The final de-noised image  $B$  can be obtained according to eq.(13).

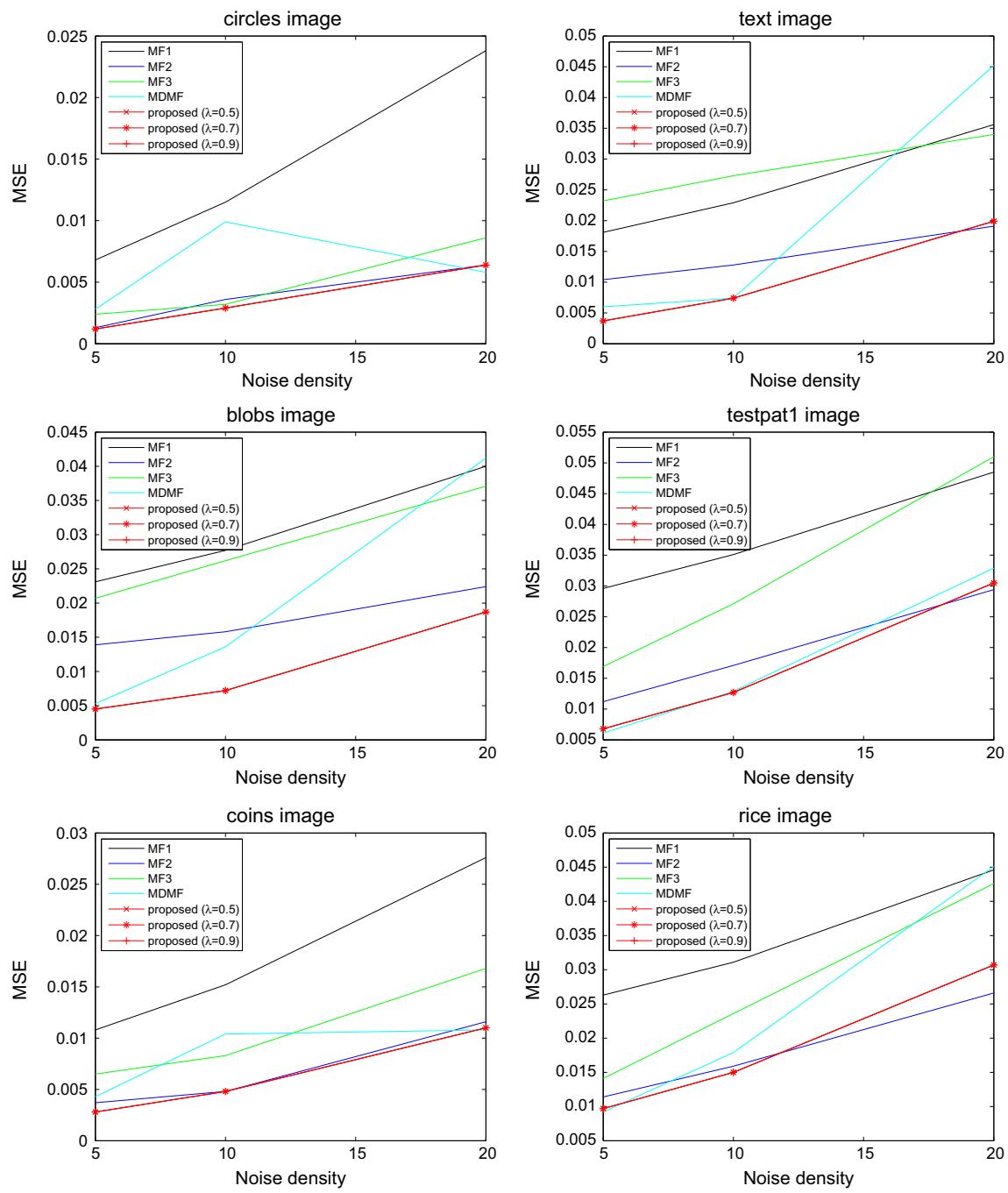
---

**Table 1** The comparison of PSNR (dB) using existing and proposed methods, and the best results were marked bold

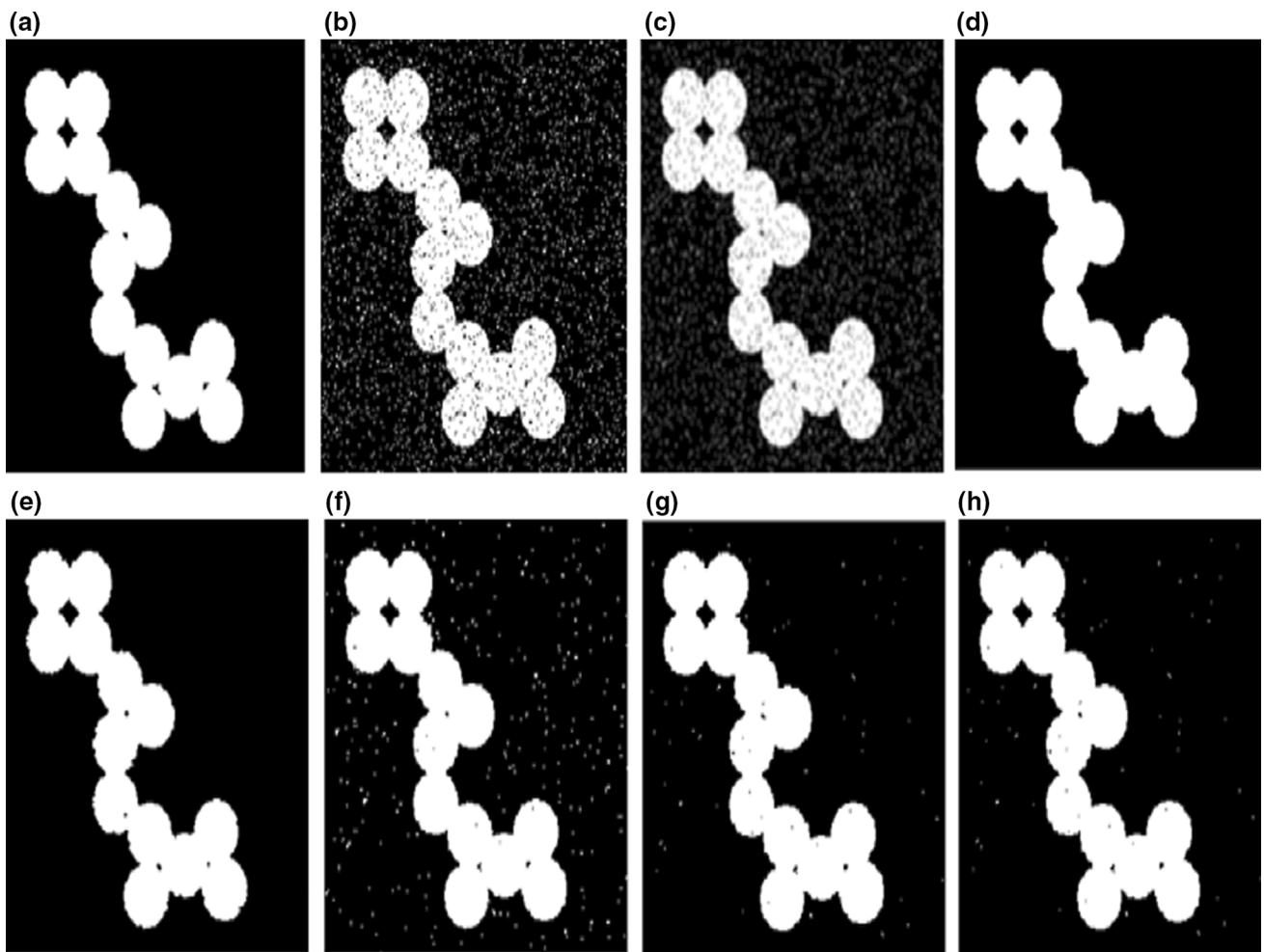
Noise density	Name	MF1	MF2	MF3	MDMF [31]	$\lambda = 0.5$	$\lambda = 0.7$	$\lambda = 0.9$
5 %	Circles	69.7935	76.9004	74.3923	73.6005	<b>77.3193</b>	<b>77.3193</b>	<b>77.3193</b>
	Text	65.5567	67.9578	64.4829	70.3296	<b>72.4395</b>	<b>72.4395</b>	<b>72.4395</b>
	Blobs	64.5015	66.7037	64.9767	70.8815	<b>71.5632</b>	<b>71.5632</b>	<b>71.5632</b>
	Testpat1	63.4125	67.6209	65.8463	<b>70.2750</b>	69.8120	69.8120	69.8120
	Coins	67.7946	72.4023	69.9899	71.7599	<b>73.7151</b>	<b>73.7151</b>	<b>73.7151</b>
	Rice	63.9265	67.5624	66.6248	<b>68.5141</b>	68.2816	68.2816	68.2816
10 %	Circles	67.5342	72.5665	73.0941	68.1598	<b>73.4626</b>	<b>73.4626</b>	<b>73.4626</b>
	Text	64.5336	67.0528	63.7646	66.6624	<b>69.4203</b>	<b>69.4203</b>	<b>69.4203</b>
	Blobs	63.7116	66.1409	63.9470	66.7920	<b>69.5663</b>	<b>69.5663</b>	<b>69.5663</b>
	Testpat1	62.6793	65.8034	63.7987	67.0270	<b>67.0892</b>	<b>67.0892</b>	<b>67.0892</b>
	Coins	66.2993	<b>71.3583</b>	68.9652	67.9634	71.3336	71.3336	71.3336
	Rice	63.1965	66.1169	64.4007	65.6100	<b>66.3568</b>	<b>66.3568</b>	<b>66.3568</b>
20 %	Circles	64.3700	70.0425	68.7982	<b>70.4978</b>	70.0528	70.0528	70.0528
	Text	62.6151	<b>65.3300</b>	62.8184	61.5827	65.1328	65.1328	65.1328
	Blobs	62.1138	64.6295	62.4397	65.4109	<b>67.6659</b>	<b>67.6659</b>	<b>67.6659</b>
	Testpat1	61.2699	<b>63.4400</b>	61.0529	62.9631	63.2918	63.2918	63.2918
	Coins	63.7273	67.4866	65.8702	<b>67.7805</b>	67.7212	67.7212	67.7212
	Rice	61.6386	<b>63.8876</b>	61.8333	61.5856	63.2636	63.2636	63.2636

**Table 2** The comparison of MSE using existing and proposed methods, and the best results were marked bold

Noise density	Name	MF1	MF2	MF3	MDMF[31]	$\lambda = 0.5$	$\lambda = 0.7$	$\lambda = 0.9$
5 %	Circles	0.0068	0.0013	0.0024	0.0028	<b>0.0012</b>	<b>0.0012</b>	<b>0.0012</b>
	Text	0.0181	0.0104	0.0232	0.0060	<b>0.0037</b>	<b>0.0037</b>	<b>0.0037</b>
	Blobs	0.0231	0.0139	0.0207	0.0053	<b>0.0045</b>	<b>0.0045</b>	<b>0.0045</b>
	Testpat1	0.0296	0.0112	0.0169	<b>0.0061</b>	0.0068	0.0068	0.0068
	Coins	0.0108	0.0037	0.0065	0.0043	<b>0.0028</b>	<b>0.0028</b>	<b>0.0028</b>
	Rice	0.0263	0.0114	0.0141	<b>0.0092</b>	0.0097	0.0097	0.0097
10 %	Circles	0.0115	0.0036	0.0032	0.0099	<b>0.0029</b>	<b>0.0029</b>	<b>0.0029</b>
	Text	0.0229	0.0128	0.0273	0.0074	<b>0.0074</b>	<b>0.0074</b>	<b>0.0074</b>
	Blobs	0.0277	0.0158	0.0262	0.0136	<b>0.0072</b>	<b>0.0072</b>	<b>0.0072</b>
	Testpat1	0.0351	0.0171	0.0271	0.0129	<b>0.0127</b>	<b>0.0127</b>	<b>0.0127</b>
	Coins	0.0152	<b>0.0048</b>	0.0083	0.0104	<b>0.0048</b>	<b>0.0048</b>	<b>0.0048</b>
	Rice	0.0311	0.0159	0.0236	0.0179	<b>0.0150</b>	<b>0.0150</b>	<b>0.0150</b>
20 %	Circles	0.0238	0.0064	0.0086	<b>0.0058</b>	0.0064	0.0064	0.0064
	Text	0.0356	<b>0.0191</b>	0.0340	0.0452	0.0199	0.0199	0.0199
	Blobs	0.0400	0.0224	0.0371	0.0412	<b>0.0187</b>	<b>0.0187</b>	<b>0.0187</b>
	Testpat1	0.0485	<b>0.0294</b>	0.0510	0.0329	0.0305	0.0305	0.0305
	Coins	0.0276	0.0116	0.0168	<b>0.0108</b>	0.0110	0.0110	0.0110
	Rice	0.0446	<b>0.0266</b>	0.0426	0.0451	0.0307	0.0307	0.0307



**Fig. 3** MSE for different methods with six test images



**Fig. 4** The restoration results of different methods (circles image): **a** original binary image, **b** 10 % salt and pepper corrupted image, **c** MF1, **d** MF2, **e** MF3, **f** MDMF [31], **g**  $\lambda = 0.5$  and **h**  $\lambda = 0.9$

### 3 Experimental results

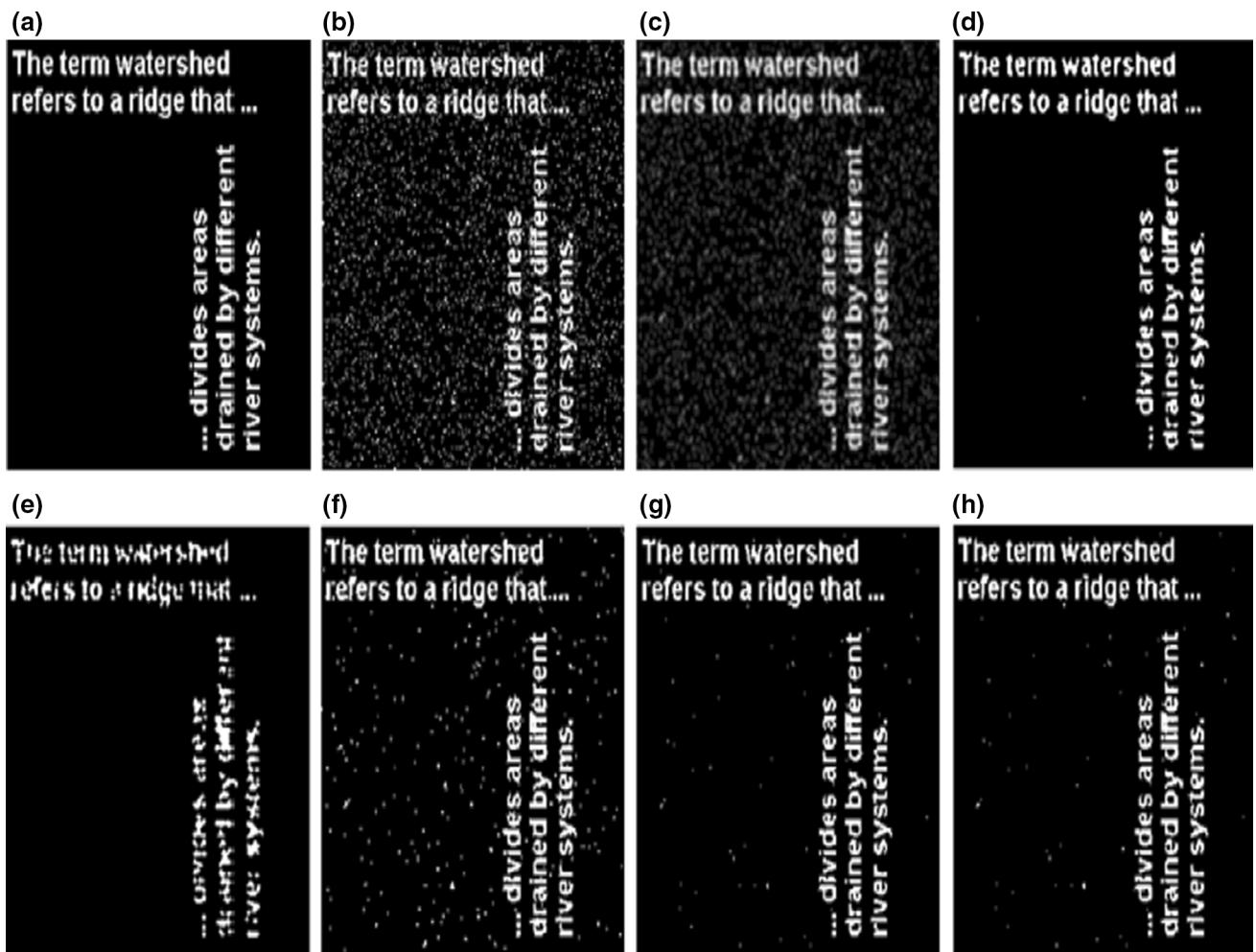
A binary image with  $10 \times 10$  pixels can be used to show the performance of proposed method (Fig. 1). In addition, the circles, text, blobs, testpat1, coins, and rice are shown in Fig. 2 to test the performance with proposed method and many existing approaches. All the experiments are carried out on a PC MATLAB 2010b with 2.40 GHZ CPU, 3GB RAM. The proposed method and mean filter, median filter, morphological filter, and MDMF [31] were evaluated

according to PSNR and MSE. The PSNR and MSE are defined as follows:

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \quad (14)$$

$$\text{MSE} = \frac{\sum_{i=1}^m \sum_{j=1}^n [Z(i,j) - A(i,j)]^2}{m \times n} \quad (15)$$

where  $Z(i, j)$  and  $A(i, j)$  denote the de-noised image and original image, respectively. As we all know, for the better



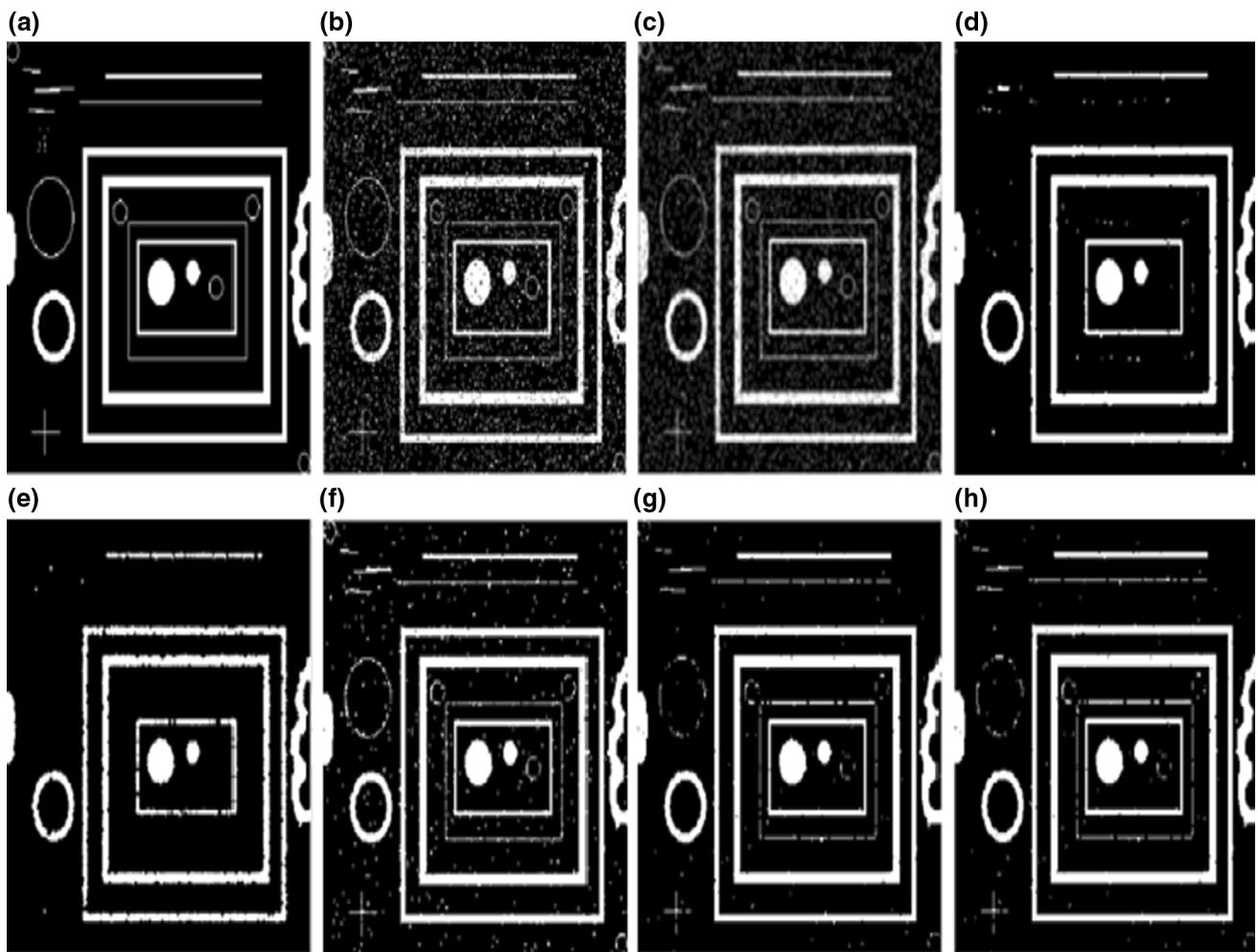
**Fig. 5** The restoration results of different methods (text image): **a** original binary image, **b** 10 % salt and pepper corrupted image, **c** MF1, **d** MF2, **e** MF3, **f** MDMF [31], **g**  $\lambda = 0.5$  and **h**  $\lambda = 0.9$

noise removal performance, the values of MSE should be very small and PSNR should be as high as possible.

Tables 1, 2 and Fig. 3 show the PSNR, MSE, and graphical illustrations results with different methods, respectively. For convenience, we use MF1, MF2, and MF3 denote the mean filter, median filter, and morphological filter, respectively. From Tables 1, 2 and Fig. 3, we can clearly see that the proposed method outperforms other existing methods in terms of PSNR and MSE values

in most cases. In addition, it must be noted that the results are the same with different  $\lambda$ . But the results are different when  $\lambda$ , that is to say, MDMF method [31]. The main reason is that the result of  $\lambda$  operate is not exceed specified threshold. In other words, the performance of proposed method is stable when  $\lambda \in [0.5, 1]$ .

Figures 4, 5, 6, 7, 8 and 9 show the visual performance of all the evaluated filters on test images corrupted with noise (10 % density). The results show that the proposed

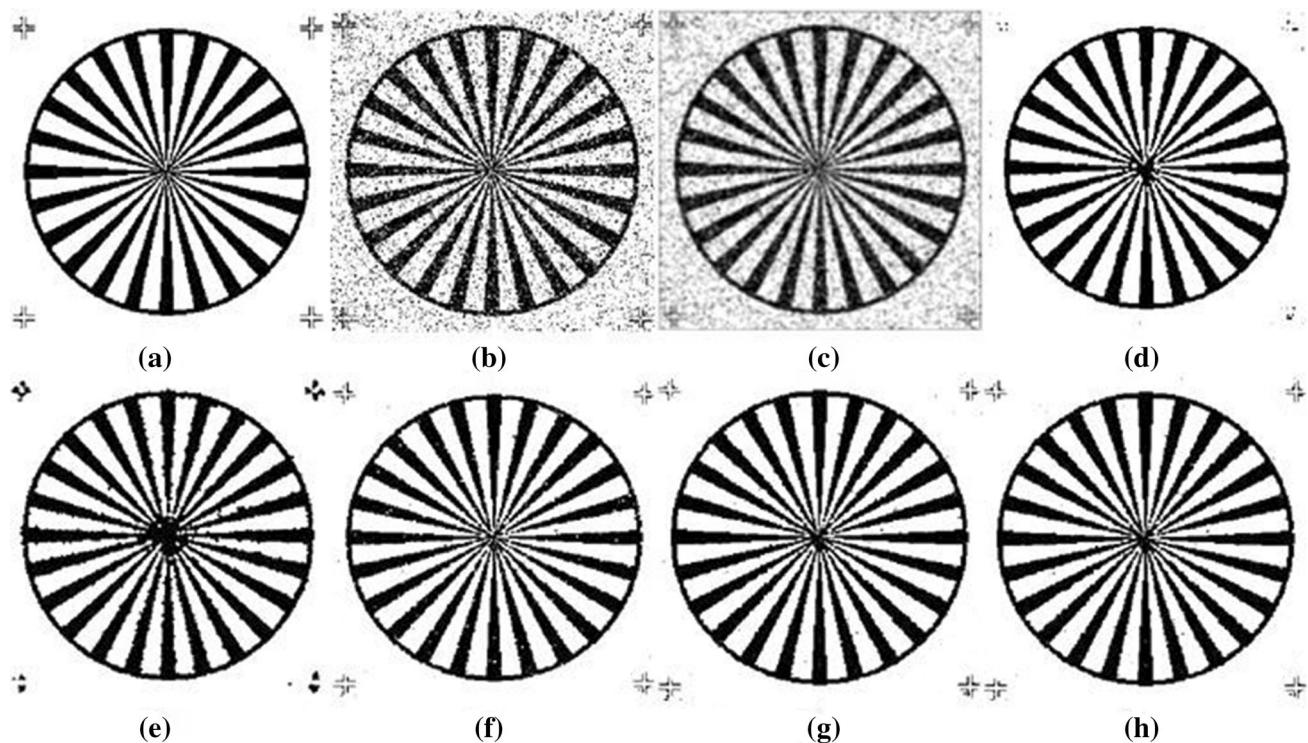


**Fig. 6** The restoration results of different methods (blobs image): **a** original binary image, **b** 10 % salt and pepper corrupted image, **c** MF1, **d** MF2, **e** MF3, **f** MDMF [31], **g**  $\lambda = 0.5$  and **h**  $\lambda = 0.9$

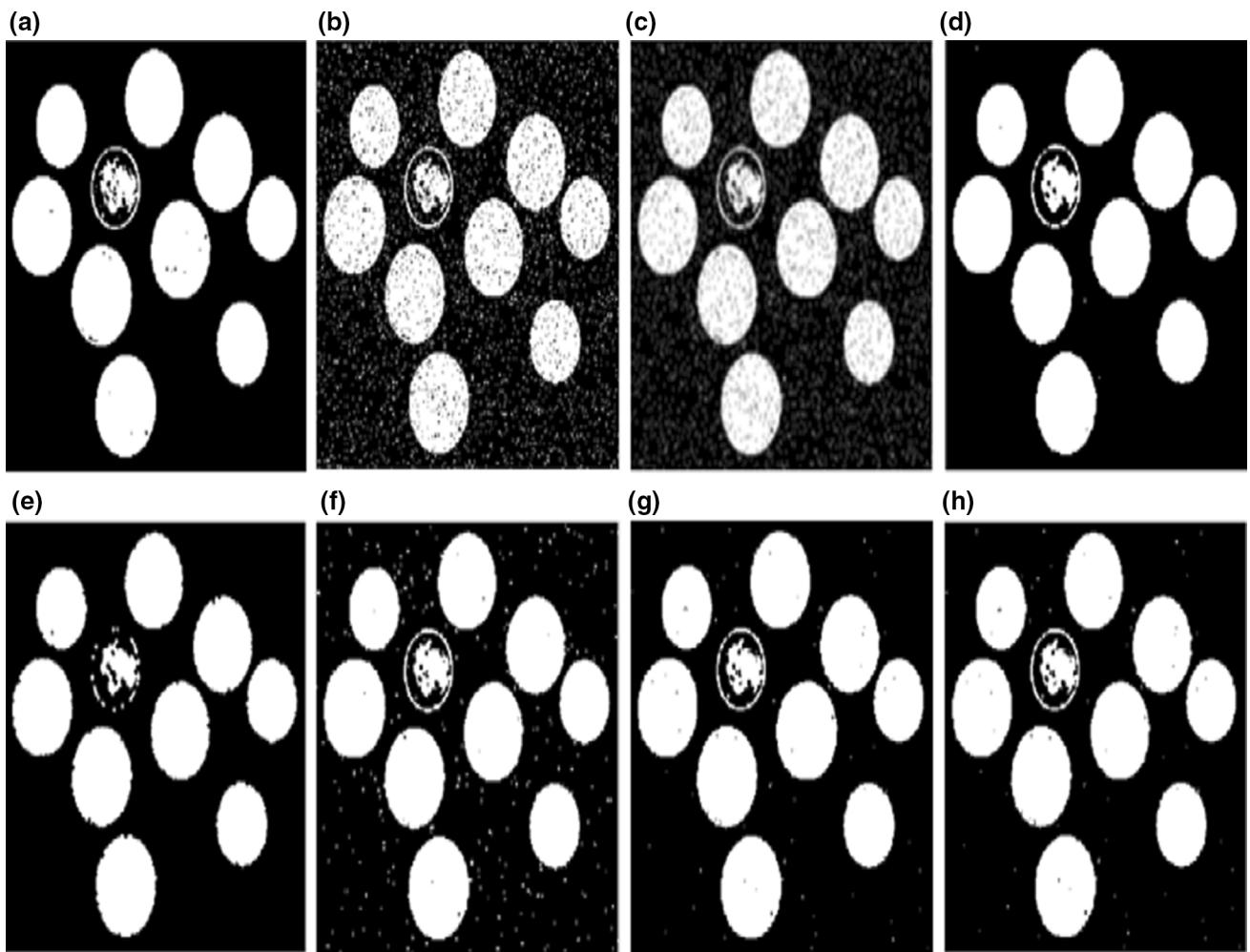
filtering technique clears the noise while preserving the shape information. On the other hand, the median filter clears the noise; however, the shape information of the image is distorted. And the morphological filter using opening followed by closing with  $3 \times 3$  disc structuring element, not only loses the shape information but also fails to remove the noise in most cases.

#### 4 Conclusion

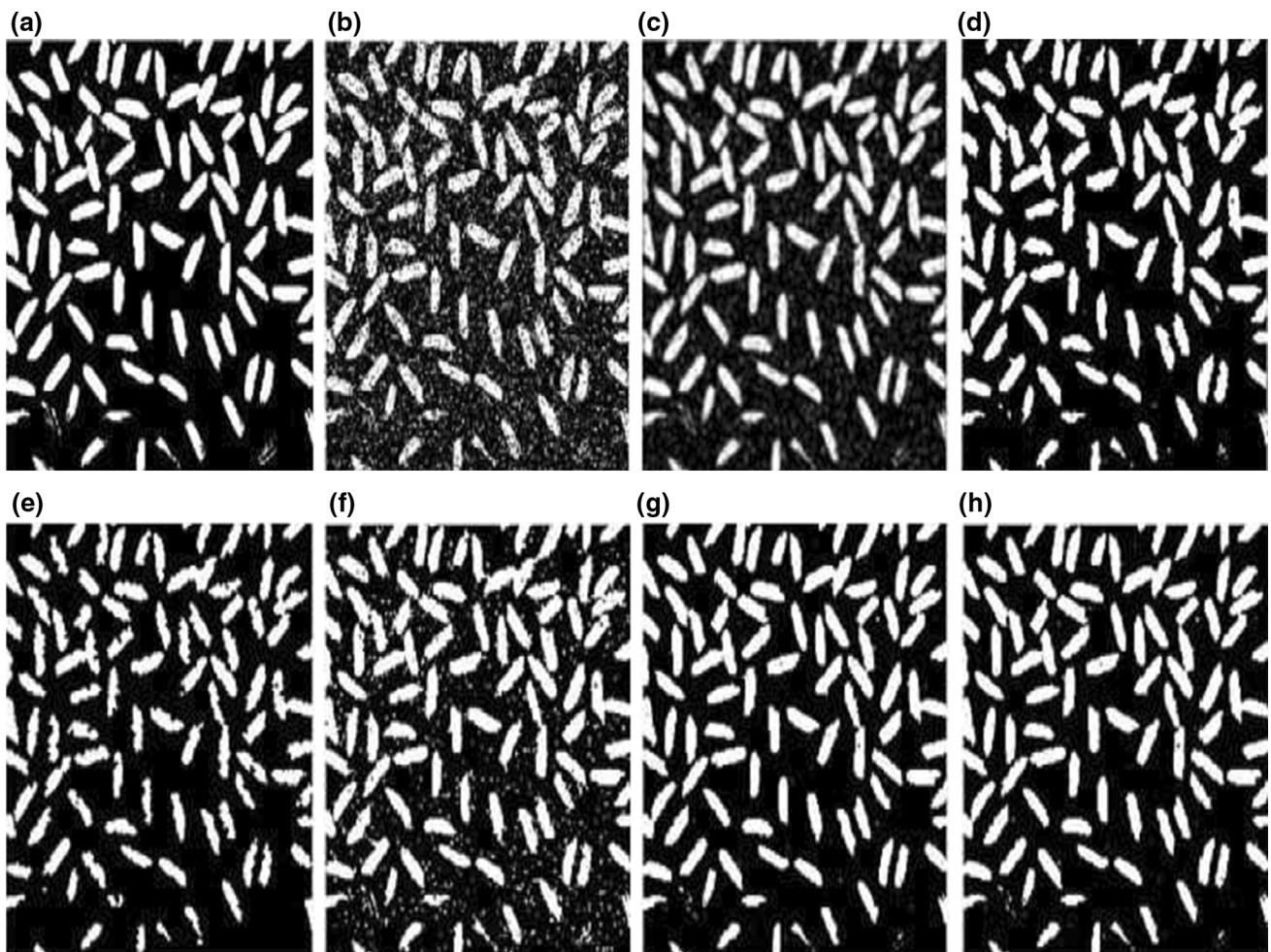
In this paper, an effective filtering method was proposed to remove the low-density noise in binary image. This method is an improved model of MDMF method. In other words, this proposed method adds one parameter by user input. This technique is based on the multiplication of  $\lambda$ -MDBM



**Fig. 7** The restoration results of different methods (testpat1 image): **a** original binary image, **b** 10 % salt and pepper corrupted image, **c** MF1, **d** MF2, **e** MF3, **f** MDMF [31], **g**  $\lambda = 0.5$  and **h**  $\lambda = 0.9$



**Fig. 8** The restoration results of different methods (coins image): **a** original binary image, **b** 10 % salt and pepper corrupted image, **c** MF1, **d** MF2, **e** MF3, **f** MDMF [31], **g**  $\lambda = 0.5$  and **h**  $\lambda = 0.9$



**Fig. 9** The restoration results of different methods (rice image): **a** original binary image, **b** 10 % salt and pepper corrupted image, **c** MF1, **d** MF2, **e** MF3, **f** MDMF [31], **g**  $\lambda = 0.5$  and **h**  $\lambda = 0.9$

with the noisy binary image followed by thresholding. Experimental results show that the proposed method outperforms some existing methods for binary low-density salt and pepper noise, both in PSNR and MSE.

**Acknowledgments** The authors would like to thank the anonymous referees and the editor for their valuable opinions. And this work is partially supported by the 111 Project under Grant No. B12018, PAPD of Jiangsu Higher Education Institutions, and the Graduates' Research Innovation Program of Higher Education of Jiangsu Province of China under Grant No. CXZZ13-0239.

#### Compliance with ethical standards

**Conflict of interest** The authors declare that they have no competing interests.

#### References

- Demirel H, Anbarjafari G (2008) Pose invariant face recognition using probability distribution functions in different color channels. *IEEE Signal Process Lett* 15:537–540
- Gao Q, Zhang L, Zhang D (2008) Face recognition using FLDA with single training image per person. *Appl Math Comput* 205(2):726–734
- Koc M, Barkana A (2011) A new solution to one sample problem in face recognition using FLDA. *Appl Math Comput* 217(24):10368–10376
- Tan X, Chen S, Zhou ZH et al (2006) Face recognition from a single image per person: a survey. *Pattern Recogn* 39(9):1725–1745
- Gao J, Fan L (2011) Kernel-based weighted discriminant analysis with QR decomposition and its application face recognition. *WSEAS Trans Math* 10(10):358–367
- Li X, Fei S, Zhang T (2009) Median MSD-based method for face recognition. *Neurocomputing* 72(16):3930–3934
- Gao J, Fan L, Xu L (2012) Solving the face recognition problem using QR factorization. *WSEAS Trans Math* 11(1):728–737
- Li X, Fei S, Zhang T (2011) Weighted maximum scatter difference based feature extraction and its application to face recognition. *Mach Vis Appl* 22(3):591–595
- Gao J, Fan L, Xu L (2013) Median null (Sw)-based method for face feature recognition. *Appl Math Comput* 219(12):6410–6419
- Gao J, Fan L, Li L et al (2013) A practical application of kernel-based fuzzy discriminant analysis. *Int J Appl Math Comput Sci* 23(4):887–903

11. Li L, Gao J, Ge H (2016) A new face recognition method via semi-discrete decomposition for one sample problem. *Optik-Int J Light Electron Opt* 127(19):7408–7417
12. Li L, Ge H, Gao J (2016) Maximum–minimum–median average MSD-based approach for face recognition. *AEU-Int J Electron Commun* 70(7):920–927
13. Han M, Sethi A, Hua W et al (2004) A detection-based multiple object tracking method. In: 2004 international conference on image processing, ICIP'04, vol 5. IEEE, pp 3065–3068
14. Bhaskar H, Dwivedi K, Dogra DP et al (2015) Autonomous detection and tracking under illumination changes, occlusions and moving camera. *Sig Process* 117:343–354
15. Karami AH, Hasanzadeh M, Kasaei S (2015) Online adaptive motion model-based target tracking using local search algorithm. *Eng Appl Artif Intell* 37:307–318
16. Yu H, Siddharth N, Barbu A et al (2015) A compositional framework for grounding language inference, generation, and acquisition in video. *J Artif Intell Res* 601–713
17. Iglesias JE, Sabuncu MR (2015) Multi-atlas segmentation of biomedical images: a survey. *Med Image Anal* 24(1):205–219
18. Nirmala S, Palanisamy V (2011) Clinical decision support system for early prediction of Down syndrome fetus using sonogram images. *SIViP* 5(2):245–255
19. Sprindzuk M, Dmitruk A, Kovalev V et al (2011) Computer-aided image processing of angiogenic histological samples in ovarian cancer. *J Clin Med Res* 1(5):249–261
20. Zuo J, Zhao C, Pan Q et al (2006) A novel binary image filtering algorithm based on information entropy. In: The sixth world congress on intelligent control and automation, 2006. WCICA 2006. IEEE vol 2, pp 10375–10379
21. Shao M, Barner KE (2006) Optimization of partition-based weighted sum filters and their application to image denoising. *IEEE Trans Image Process* 15(7):1900–1915
22. Maggioni M, Katkovnik V, Egiazarian K et al (2013) Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Trans Image Process* 22(1):119–133
23. Zhang W, Yu F, Guo H (2009) Improved adaptive wavelet threshold for imagedenoising. In: Chinese control and decision conference, 2009. CCDC'09. IEEE, pp 5958–5963
24. Xie ZJ, Song BY, Zhang Y et al (2014) Application of an improved wavelet threshold denoising method for vibration signal processing. *Adv Mater Res* 889:799–806
25. Yan R, Shao L, Liu Y (2013) Nonlocal hierarchical dictionary learning using wavelets for image denoising. *IEEE Trans Image Process* 22(12):4689–4698
26. Zhong JJ, Fang SN, Linghu CY (2014) Research on application of wavelet denoising method based on signal to noise ratio in the bench test. *Appl Mech Mater* 457:1156–1162
27. Salimi-Khorshidi G, Douaud G, Beckmann CF et al (2014) Automatic denoising of functional MRI data: combining independent component analysis and hierarchical fusion of classifiers. *Neuroimage* 90:449–468
28. Ibrahim Z, Khalid N K, Ibrahim I et al (2008) A noise eliminationprocedure for printed circuit board inspection system. In: Second Asia international conference on modeling and simulation, 2008. AICMS 08. IEEE, pp 332–337
29. Graf A, Smola AJ, Borer S (2003) Classification in a normalized feature space using support vector machines. *IEEE Trans Neural Netw* 14(3):597–605
30. Liu X, Zhai D, Zhao D et al (2014) Progressive image denoising through hybrid graph Laplacian regularization: a unified framework. *IEEE Trans Image Process* 23(4):1491–1503
31. Anbarjafari G, Demirel H, Gokus AE (2014) A novel multi-diagonal matrix filter for binary image denoising. *J Adv Electr Comput Eng* 1(1):14–21