IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 June 2022, accepted 12 July 2022, date of publication 15 July 2022, date of current version 20 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3191335

RESEARCH ARTICLE

An Efficient Gaussian Filter Based on Gaussian Symmetric Markov Random Field

FUSONG XIONG[1,2], JIAN ZHANG[1,3], ZHIQIANG ZHANG[1,2], AND YUN LING[1,2]

[1] Soochow College, Soochow University, Suzhou, Jiangsu 215006, China
[2] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, Nanjing, Jiangsu 210008, China
[3] Wenzheng College, Soochow University, Suzhou, Jiangsu 215104, China

Corresponding author: Fusong Xiong (xiongfusong@suda.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672369, Grant 6177255, Grant 62072321, and Grant 61972454; in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization; and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions, China.

ABSTRACT This article presents a new image denoising algorithm that uses Gaussian Symmetric Markov random fields based on maximum a posteriori estimation. First, an image denoising model based on Gaussian Symmetric Markov random fields is built, and the image denoising problem was converted to a maximum a posteriori estimation problem. The prior probability of an image can be estimated using the Gibbs distribution, which is equivalent to Markov random fields. Second, the maximum a posteriori estimation is calculated using the expectation-maximization algorithm and conjugate gradient method, where the expectation-maximization algorithm is used to estimate Gaussian Symmetric Markov random field hyper-parameters and the conjugate gradient method is used to calculate the criterion function. The experimental results for the synthetic images and the standard Berkeley segmentation datasets demonstrate the success of the proposed Gaussian denoising filter, as compared with the state-of-the-art methods such as BM3D, WNNM, SGWD-HMMs, SSLBD, DnCNN and BUIFD.

INDEX TERMS Gaussian Markov random fields, expectation-maximization, conjugate gradient, denoising filter.

I. INTRODUCTION

In the process of image acquisition, encoding, storage and transmission, all the images are ‘‘dirtied’’ by visible or invisible noise that may degrade the image [1], [2]. The image noise mostly includes impulse (salt and pepper) noise, Gaussian noise and mixed Noise etc. [2], [3]. Noise not only affects the visual comprehension of the image but also hinders the normal recognition. Therefore, the image denoising filter is an important task in image processing.

An image containing Gaussian noise actually contains uncertain gray values that vary within a certain range based on the gray value of each pixel of the original image [4], [5]. The essential problem of image denoising is to make the noisy image as close as possible to the original image after denoising. Gaussian noise, represented by a Gaussian distribution function, is additive and independent, and is caused by three common factors: amplifier noise, shot noise, and grain noise of film [6], [7]. Accordingly, the noisy image can be expressed as:

$$Y_n(i,j) = X(i,j) + n(i,j) \tag{1}$$

where X is the original image, n is the additive noise on the pixel basis, and Y represents the noisy image of X.

In the past few decades, a variety of models have been successfully used for image denoising, including sparse model [8], [9], gradient model [10], [11], Markov random fields (MRF) model [12], [13] and Hidden Markov Model (HMM) model [14], [15], non-local self-similarity model [16], [17], convolutional neural network based models [18], [19] and deep learning based model [20]. In addition, some filter-based image denoising methods [21]–[23] have also been successfully proposed. Semi-supervised [24] and self-supervised [25] denoising algorithms are the other two effective denoising algorithms.

The associate editor coordinating the review of this manuscript and approving it for publication was Gerardo Di Martino.

Among the above methods, there are many very popular ones. Such as, BM3D [16] and WNNM [17], which are nonlearning-based and use local or non-local structures of an input noisy image. Recently, convolutional neural network (CNN) based methods are proved to be effective denoising algorithms, such as IRCNN [18], DnCNN [19]. BUIFD [20] is another deep learning based image denoiser, and it claims to be better than the DnCNN. Yin *et al.* [25] proposed a new image reconstruction model for the impulse noise removal problem.

The existing denoising filters mostly focus on edge and structure preservation [7]. Although their high denoising quality, these methods usually involve high complexity of computation, manually chosen parameters, and large amounts of training data, etc.

Markov Random Field (MRF) was introduced into the image field by Besag in 1974, and has been widely used in image segmentation, classification, and restoration [26]. MRF is essentially a conditional probability model [27], combined with the Bayesian criterion, the problem can be reduced to solve the model's maximum a posterior probability problem. In this paper, Gaussian Symmetric Markov random field (GSMRF) is applied to the Gaussian denoising. The image noise degradation model based on GSMRF is first established, and then the hyper-parameters estimation of a prior probability model of the image is performed using the Expectation-Maximization (EM) algorithm. Finally, the conjugate gradient method is used to minimize the criterion function to obtain the denoised image. Comparing with the state-of-the-art methods such as BM3D [16], WNNM [17], SGWD-HMMs [14], SSLBD [25], DnCNN [19], and BUIFD [20], the performance of our Gaussian denoiser, as indicated by the experimental results, is particularly attractive in Gaussian denoising quality.

The contributions of this paper are:

(1) We propose a novel noise degradation model based on Gaussian Symmetric Markov random field, which can convert the problem to maximum a posteriori estimation.

(2) We use EM algorithm to estimate the hyper-parameters automatically, not manually chosen parameters, the conjugate gradient algorithm to optimize the criterion function. The convergence performance of the two algorithms can be guaranteed theoretically.

(3) Our method outperforms the state-of-the-art denoising methods both on synthetic datasets and real-world datasets, which shows its potential applications in real-world scenarios.

The remainder of this paper is organized as follows. In Section 2, the image degradation (noise) model based on GMRF is established. In Section 3, we give the method of hyper-parameter estimation. In Section 4, the image denoising algorithm is proposed. In Section 5, the experimental results are presented and discussed. The final section concludes the paper.

II. IMAGE DEGRADATION MODEL BASED ON GAUSSIAN SYMMETRIC MARKOV FIELD

Assume an image is defined in a two-dimensional space $L = \{(i,j)\,|i \in \{1,2,\ldots,m\}, j \in \{1,2,\ldots,n\}\}, N = m \times n$ is the number of pixels. The status of each pixel can be represented by the random variables $X_i\ (i \in L)$. All the $m \times n$ random variables constitute a discrete random field $X = (X_1, X_2 \ldots, X_N), N = m \times n$. A digital image can be seen as an implementation of the random field X. In other words, an image is a configuration of X, which represent as: $x = (x_1, x_2 \ldots, x_N)$. All the possible configurations make up a space Ω. Assume η_i be the set of neighbors located at the point x_i. η_i consisted of the nearest 4 or 8 pixels, which means first order neighborhood system or second order neighborhood system (as shown in Fig. 1(a) and (b)), respectively.

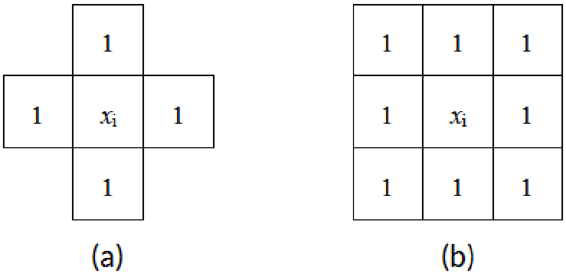


FIGURE 1. **(a) First order neighborhood system, (b) Second order neighborhood system.**

If the following conditions are satisfied:

$$P(X = x) > 0, \quad \forall x \in \Omega \tag{2}$$

$$\begin{aligned} &P(X_i = x_i | X_l = x_l, l \in \{1,2,\ldots,N\}, l \neq i) \\ &= P(X_i = x_i | X_l = x_l, l \in \eta_i) \end{aligned} \tag{3}$$

Then, the X is called a Markov random field (MRF) with $\eta = \{\eta_i, i \in \{1,2,\ldots,N\}\}$ as the neighborhood system. It means that the characteristic of a pixel is completely determined by its neighborhood system η_i. From the equation (3), it can be seen that MRF can be described by a conditional distribution, and this distribution is a local characteristic of a random field. The Hammersley-Clifford theorem [28] points out that each MRF corresponds with a Gibbs random field (GRF). The GRF describes the global characteristics of the random field. The MRF can be determined if the energy function of the GRF is given.

Image restoration methods based on the MRF are usually based on Bayes' theorem. Assume X is the original image, and Y is the degraded image of X, we have:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \propto P(Y|X)P(X) \tag{4}$$

where the $P(X|Y)$ and $P(X)$ are the a posteriori and a priori probability density function of the unknown, respectively, and $P(Y)$ is a constant. The image restoration can be attributed to solving the problem of maximum a posterior probability (MAP), as follows:

$$\hat{X}_{MAP} = \underset{x}{arg\,max}\, P\left(Y|X\right) P\left(X\right) \tag{5}$$

From the Hammersley - Clifford theorem, it is easy to know that the joint probability density function of MRF can describe by Gibbs distribution, and its expression is shown as follows:

$$P\left(X;\omega\right) = \frac{1}{Z\left(\omega\right)} exp\left\{-\varepsilon\left(x;\omega\right)\right\} \tag{6}$$

where Z is the normalized constant (also called the partition function) which normalizes integral of $P\left(X;\omega\right)$ to 1, $\varepsilon\left(\cdot\right)$ denotes MRF's energy function, and ω is called hyper-parameter.

In order to find the solution, we select a Gaussian Symmetric Markov random field [29], [30], because it can fuse the contextual information of the image very well [31]. In this paper, the following energy function for $\varepsilon\left(\cdot\right)$ is taken:

$$\begin{aligned}\varepsilon\left(x;\omega\right) &= \sum_{i=1}^{N}\sum_{j\in\eta_i}\left[\frac{1}{2\left(\omega_{ij}\right)^2}\left(x_i-x_j\right)^2\right], \omega \\ &= \left\{\omega_{ij}, i\in\left\{1,2,\ldots,N\right\}\right\}, \quad j\in\eta_i\end{aligned} \tag{7}$$

where η_i represents the second order neighborhood system of the ith pixel in the image (as shown in Fig. 1(b)), N is the number of pixels in the image, and $\frac{1}{\omega_{\mathrm{ij}}^2}\left(\mathrm{x_i}-\mathrm{x_j}\right)^2$ is the clique potential, which model the strength of the interaction between *ith* and *jth* pixels. ω_{ij} usually depends on the direction between ith and *jth* pixels. We select a symmetric markov random field, where $\omega_{ij}=\omega_{ji}$.

A common reason for image degradation is due to Gaussian noise (the problem we focus on in this article). We assume that the image is noised by Gaussian noise constituted by i.i.d. random zero-mean Gaussian variables with variance σ_n^2. In such case, $P\left(Y|X\right)$ can be described approximately with a Gaussian distribution:

$$P\left(Y|X\right) \propto exp\left(-\frac{\|y-x\|^2}{2\sigma_n^2}\right) \tag{8}$$

By substituting (6), (7) and (8) into (5), we have:

$$\hat{X}_{MAP} = \underset{x}{arg\,min}\left[\frac{\|y-x\|^2}{2\sigma_n^2}+\varepsilon\left(x;\omega\right)\right] \tag{9}$$

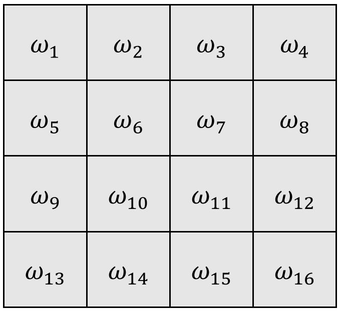
Now, the denoising problem becomes to find the solution of equation (9). However, ω and x are unknown. In order to implement (9), we must to estimate the values of the hyper-parameters ω and the image x. The image x is available data sets, referred as the incomplete and complete date corresponding to the real and synthetic images, respectively.

III. HYPER-PARAMETER ESTIMATION

For the hyper-parameters ω, can be estimated through the noisy image Y. We consider a Symmetric MRF ($\omega_{ij}=\omega_{ji}$), it needs to estimate $4N$ hyper-parameters (each pixel has four parameters). Therefore, it's very difficult to estimate all the hyper-parameters in equation (7) at the same time. Thus, some methods have to be found to overcome this difficult. Here we divide the image X in G domains, so that all parts $G_i, i=1,2,\ldots,G$ cover the entire image without overlapping each other. For example, divide the image into 4×4 parts as shown in Fig. 2. Assuming that the statistical properties of each part are described by one hyper-parameter and each part is independent of each other parts. In this case:

$$\begin{aligned}P\left(X;\omega\right) &= \prod_{g=1}^{G} P\left(X_g;\omega_g\right) \\ &= \prod_{g=1}^{G}\frac{1}{Z_g} exp\left\{-\frac{U\left(x_g\right)}{2\left(\omega_g\right)^2}\right\}\end{aligned} \tag{10}$$

where $U\left(\varphi\right) = \sum_{i\in gthpart}\sum_{j\in\eta_i}\left(\varphi_i-\varphi_j\right)^2$, $x_g = \left[x_{g,1},x_{g,2},\ldots,x_{g,N_g}\right]$, N_g is the total pixels of the gth partition image, and $N = \sum_{g=1}^{G} N_g$, ω is the accordingly hyper-parameter, and $\omega = \left[\omega_1,\omega_2,\ldots,\omega_g\right]^T$ is the hyper-parameter vector. It shows from [31] that $Z_g = \left(\omega_g\right)^{N_g} z$, and z is a constant factors independent on ω.



ω_1	ω_2	ω_3	ω_4
ω_5	ω_6	ω_7	ω_8
ω_9	ω_{10}	ω_{11}	ω_{12}
ω_{13}	ω_{14}	ω_{15}	ω_{16}

FIGURE 2. **4 × 4 image partition, 16 hyper-parameters.**

Through the above analysis and process, the hyper-parameters $\omega = \left[\omega_1,\omega_2,\ldots,\omega_{16}\right]^T$, only 16 hyper-parameters need to be estimated. Here we choose the iterative algorithm EM (expectation–maximization), which for solve the maximum likelihood estimation of the hyper-parameters from incomplete data, and its convergence performance has been guaranteed [32], [33]. The EM algorithm repeats two steps (Step E and Step M) until convergence, as follows:

STEP E:

$$M\left[\omega,\omega\left(t\right)\right] = E\left[lnP\left(X;\omega\right)|Y=y,\omega\left(t\right)\right] \tag{11}$$

STEP M:

$$\omega\left(t+1\right) = \underset{\omega}{arg\,max}\, M\left[\omega,\omega\left(t\right)\right] \tag{12}$$

By referring to equation (10), we have:

$$\begin{aligned}&\omega\left(t+1\right) \\ &= \underset{\omega}{\mathrm{argmin}}\, E\left\{-ln\left[\prod_{g=1}^{G}\frac{1}{\left(\omega_g\right)^{N_g} z} exp\left\{-\frac{U\left(x_g\right)}{2\left(\omega_g\right)^2}\right\}\right]|Y\right.\end{aligned}$$

$$\left. = y, \omega\left(t\right) \right\} \tag{13}$$

It's solved as in [33]:

$$\left(\omega_g\right)^2 = \frac{E\left[U\left(x_g\right)|Y = y, \omega\left(t\right)\right]}{N_g} \tag{14}$$

According to the [31], the expected value of Step E should be sampled from a posterior distribution $P_{X|Y}\left(x|y\right)$ as follow:

$$\begin{aligned} & P_{X|Y}\left(x|y\right) \\ & = \frac{1}{\left(2\pi\sigma_n^2\right)^{N/2} Zf_Y\left(y\right)} exp\left\{-\frac{\|y-x\|^2}{2\sigma_n^2}\right\} \\ & \quad - \sum_{g=1}^{G} \frac{U\left(x_g\right)}{2\left(\omega_g\right)^2} \end{aligned} \tag{15}$$

Equation (15) is actually an expression of Gibbs distribution, which its energy function is the exponential. Therefore, a Gibbs-Sampler can be used to generate samples X. The Monte Carlo techniques [28] method for generating samples from a Gibbs distribution is given by [26]. However, The Monte Carlo techniques are complicated which has heavily computational task. Here, we refer to the method of [30], which is based on the assumption that the local a posterior probability distribution function is Gaussian:

$$P\left(x_i|x_j; j \in \eta_i, y\right) = \frac{1}{\sqrt{2\pi}\,\omega_i} exp\left[-\frac{\left(x_i-\mu_i\right)^2}{2\left(\omega_i\right)^2}\right] \tag{16}$$

where:

$$\mu_i = \frac{\sum_{j\in\eta_i} \frac{x_j}{2\left(\omega_{ij}\right)^2} + \sum_{h=1}^{N} \frac{y_i}{4\sigma_n^2}}{\sum_{j\in\eta_i} \frac{1}{2\omega_{ij}^2} + \sum_{h=1}^{N} \frac{1}{4\sigma_n^2}} \tag{17}$$

$$\omega_i^2 = \left[\sum_{j\in\eta_i} \frac{2}{\left(\omega_{ij}\right)^2} + \sum_{h=1}^{N} \frac{1}{\sigma_n^2}\right]^{-1} \tag{18}$$

Consequently, we can generate samples of X by using a Gaussian sampler instead of Gibbs Sampler.

IV. DENOISING ALGORITHM

The denoising algorithm is to find the minimum value of the criterion function (9). In this paper, we use the conjugate gradient method [34] to achieve minimization. This minimization process needs to estimate the gradient of the criterion function $\phi_{MAP}\left(x\right)$, which is decomposed into two parts ϕ_{ML} and ϕ_{MAF}.

$$\hat{x}_{MAP} = \underset{x}{arg\,min}\,\phi_{MAP}\left(x\right) \tag{19}$$

where,

$$\phi_{MAP}\left(x\right) = \underbrace{\frac{\|y-x\|^2}{2\sigma_n^2}}_{\phi_{ML}(x)} + \underbrace{\sum_{g=1}^{G} \frac{\sum_{i=1}^{N_g}\sum_{j\in\eta_{g,i}}\left(x_{g,i}-x_{g,j}\right)^2}{2\omega_g^2}}_{\phi_{MAF(x)}} \tag{20}$$

$$\nabla_c\phi_{ML}\left(x\right) = -\frac{\left(y-x\right)^2}{\sigma_n^2} \tag{21}$$

$$\nabla_c\phi_{MRF}\left(x\right) = \left\{ \begin{aligned} & 2\sum_{j\in\eta_k} \frac{1}{\left(\omega_{kj}\right)^2}\left(x_k-x_j\right), \\ & k = 1, 2, \ldots, N \end{aligned} \right\} \tag{22}$$

The gradient of $\phi_{MAP}\left(x\right)$ is given by:

$$\nabla_c\phi_{MAP}\left(x\right) = \nabla_c\phi_{ML}\left(x\right) + \nabla_c\phi_{MRF}\left(x\right) \tag{23}$$

It becomes easy to minimize the criterion function (9) by using the conjugate gradient method after we obtained the gradient $\nabla_c\phi_{MAP}\left(x\right)$. A description of the denoising algorithm (we call it as a GSMRF) as follows:

V. EXPERIMENTAL RESULTS

In this section, we report the experimental results obtained using the proposed method. In order to assess the effectiveness of the proposed GSMRF filter, we qualitatively and quantitatively assessed on lots of grayscale and color images. These experimental images include two synthetic images and the standard Berkeley segmentation datasets (BSD68 and CBSD68) [35], which were widely used for the evaluation of denoising methods.

The results provided by the proposed method were compared with those yielded by six state-of-the-art denoising algorithms which widely used in the literature, i.e., BM3D [16], WNNM [17], SGWD-HMMs [14], SSLBD [25], DnCNN [19] and BUIFD [20]. As the comparison methods, all the parameter values of BM3D, WNNM, SGWD-HMMs, SSLBD, DnCNN and BUIFD were set to the same as default values according to [14], [16], [17], [19], [20], and [25]. The testing codes of DnCNN and BUIFD methods are download from the author's websites[1],[2]. At present, there are several measures that may be used in the evaluation of the effectiveness of image denoising algorithms. The most widely used are Peak Signal to Noise Ratio (*PSNR*) and Structural Similarity (*SSIM*), which were employed to evaluate the performances of all the denoising methods. A high *PSNR* and *SSIM* values indicate improved image reconstruction. In addition, in all the experiments, we choose eight noise levels, i.e., the noise standard deviation $\sigma = 5, 10, 15, 25, 35, 45, 55$ *and* 65.

A. EXPERIMENT ON SYNTHETIC IMAGES

Fig. 3 shows two original images with 256×256 pixels, which named as "Squares" and "Circles" [Fig. 3(a) and 3(b)], respectively. In Fig. 3(a), we place some squares (their gray

[1] https://github.com/cszn/DnCNN

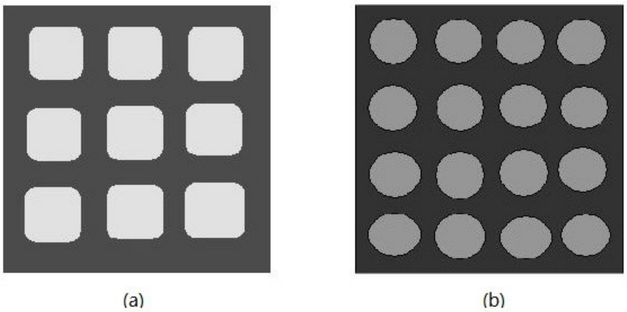
[2] https://github.com/majedelhelou/BUIFD

Procedure 1 Procedures of GSMRF

```
BEGIN
Input an image $X$ of size $m \times n$, add Gaussian noise with zero-mean and variance $\sigma_n^2$, to make it into a noisy image $Y$;
Initialization hyper-parameter $\omega(0)$ and convergence threshold $T > 0$, and $i = 0$;
Calculate $M[\omega, \omega(i)]$; ( Step E)
repeat
    $\omega(i+1) = argmax\, M[\omega, \omega(i)]$; (Step M)
    Update $M[\omega, \overset{\omega}{\omega}(i+1)]$;
    $i = i + 1$;
until $|M[\omega, \omega(i+1)] - M[\omega, \omega(i)]| \leq T$
$\hat{\omega} = \omega(i+1)$;
Initialize the original image $x^0$, threshold $\varepsilon > 0$, right step length $\rho$, and $k = 0$;
Calculate the gradient $\nabla_c \phi_{MAP}(x^0)$ at $x^0$, initialize the search direction $s_0 = -\nabla_c \phi_{MAP}(x^0)$, and normalize $s_0$: $\hat{s}_0 = \frac{-\nabla_c \phi_{MAP}(x^0)}{\|-\nabla_c \phi_{MAP}(x^0)\|}$;
repeat
    Calculate $x^{k+1} = x^k + \rho \cdot \hat{s}_k$
    Calculate $\nabla_c \phi_{MAP}(x^{k+1})$, $\nu_k = \frac{\|\nabla_c \phi_{MAP}(x^{k+1})\|^2}{\|\nabla_c \phi_{MAP}(x^k)\|^2}$, $s_{k+1} = -\nabla_c \phi_{MAP}(x^{k+1}) + \nu_k \cdot s_k$
    Normalize $\hat{s}_{k+1}$: $\hat{s}_{k+1} = \frac{s_{k+1}}{\|s_{k+1}\|}$
    Update $\phi_{MAP}(x^{k+1})$;
    $k = \mathrm{k} + 1$
until $|\phi_{MAP}(x^{k+1}) - \phi_{MAP}(x^k)| \leq \varepsilon$
$\hat{X}_{MAP} = x^{k+1}$
Output the denoised image $\hat{X}_{MAP}$.
END
```

level is 225) on the darker background (its gray level is 75). In Fig. 3(b), we place some circles (their gray level is 150) on the darker background (its gray level is 50). The Gaussian noise is added to the original image. Fig. 4(b) and Fig. 5(b) are two examples of the Gaussian noise "Squares" image with zero-mean and the corresponding noise standard deviation $\sigma = 15, 35$, respectively. Fig. 6(b) and Fig. 7(b) are two examples of the Gaussian noise "Circles" image with zero-mean and the corresponding noise standard deviation $\sigma = 15, 35$, respectively. Figs. 4-7 provide the visual comparison between the denoising results obtained by BM3D, WNNM, SGWD-HMMs, SSLBD, DnCNN, BUIFD and the proposed GSMRF.

As can be seen from Figs. 4-7, there are obvious noises in the results obtained by WNNM and SSBND no matter whether the noise level σ is 15 or 35 [see (d) and (f) of Figs. 4-7]. Although the BM3D, SGWD-HMMs, DnCNN, BUIFD, and GSMRF can all remove most of the noise from the image, the results obtained by the BM3D, SGWD-HMMs, and BUIFD usually obtain some artifacts, which cause the denoising images to look blurred [see (c), (f), and (h) of Figs. 4-7]. In contrast, DnCNN and GSMRE can obtain visually satisfactory results, especially our proposed method can yield best denoising images in all eight test cases. Reference [36] concludes that "The denoising methods based on discriminant training usually yield better results for images with irregular textures, while denoising methods based on non-local similarity usually yield better results for images with regular and repetitive structures. Our results on two synthetic images are consistent with the conclusion in [36], because the "Squares" and "Circles" are just images with regular and repetitive structures. In fact, it's a reasonable result in intuitively due to the image with regular and repetitive structures can well satisfy non-local similarity priors, which leading to better results.


(a)

(b)

FIGURE 3. Two synthetic test images. (a) Squares. (b) Circles.

Table 1 and 2 list the results in terms of *PSNR* and *SSIM* obtained by applying each denoising method under different noise levels (5, 10, 15, 25, 35, 45, 55 and 65). The average values of Peak Signal to Noise Ratio ($\bar{\varphi}_{PSNR}$) and Structural Similarity ($\bar{\varphi}_{SSIM}$) for all the eight test cases are also reported in Table 1 and 2 to evaluate the effectiveness of the seven denoising methods. The best values of *PSNR* and *SSIM* with different noise levels, of $\bar{\varphi}_{PSNR}$ and of $\bar{\varphi}_{SSIM}$ for each synthetic image are highlighted in bold.

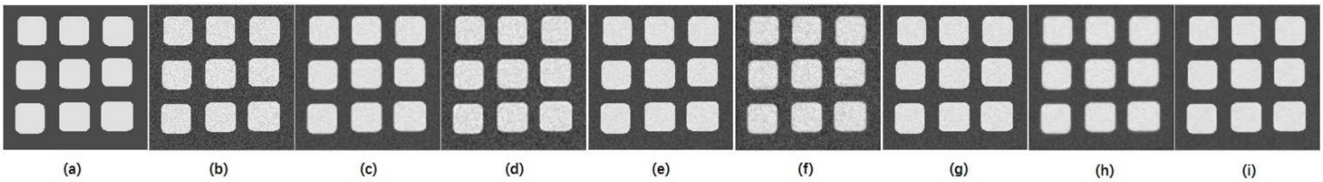


FIGURE 4. Denoising results of synthetic "Squares" image with noise level $\sigma = 15$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

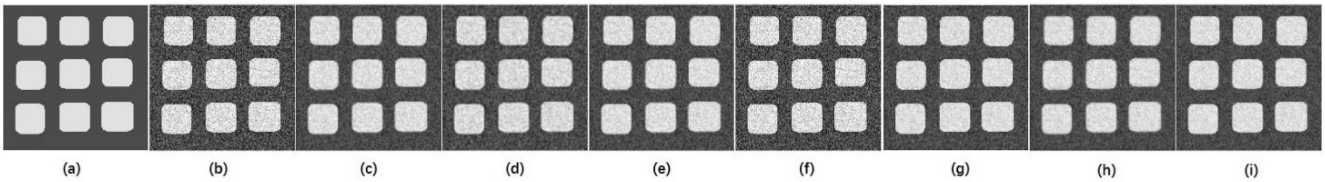


FIGURE 5. Denoising results of synthetic "Squares" image with noise level $\sigma = 35$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

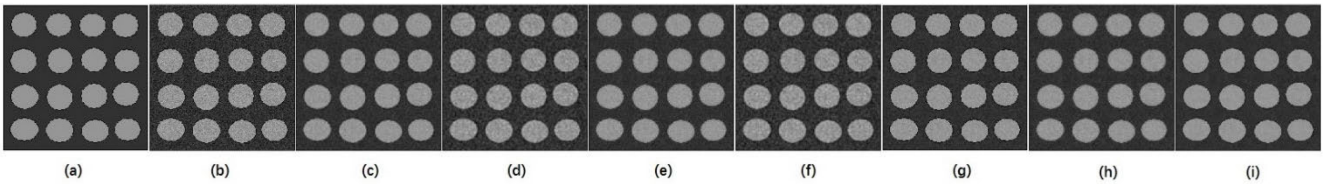


FIGURE 6. Denoising results of synthetic "Circles" image with noise level $\sigma = 15$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

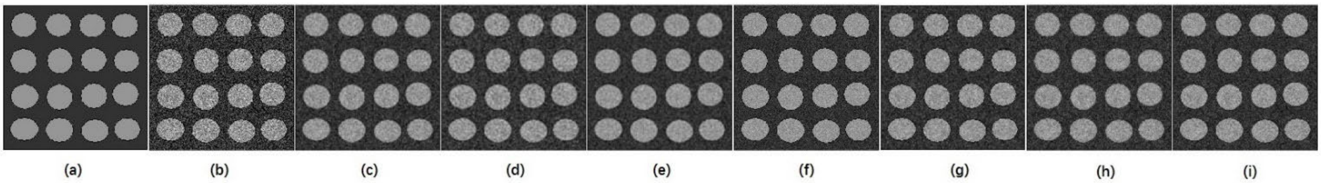


FIGURE 7. Denoising results of synthetic "Circles" image with noise level $\sigma = 35$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

As shown in Table 1 and 2, in all the test cases, the *PSNR* and *SSIM* obtained by GSMRF are the highest ones. For the "Squares", the $\bar{\varphi}_{PSNR}$ value obtain by GSMRF outperforms the BM3D, WNNM, SGWD-HMMs, SSLBD, DnCNN and BUIFD methods by 0.32dB, 0.33dB, 0.34dB, 1.73dB, 0.17dB and 0.21dB respectively, and the $\bar{\varphi}_{SSIM}$ value obtain by GSMRF outperforms the BM3D, WNNM, SGWD-HMMs, SSLBD, DnCNN and BUIFD methods by 0.29%, 0.46%, 0.42%, 2.09%, 0.08% and 0.07%, respectively. For the "Circles", the $\bar{\varphi}_{PSNR}$ value obtain by GSMRF outperforms the BM3D, WNNM, SGWD-HMMs, SSLBD, DnCNN and BUIFD methods by 0.28dB, 0.36dB, 0.72dB, 1.37dB, 0.20dB and 0.10dB respectively, and the $\bar{\varphi}_{SSIM}$ value obtain by GSMRF outperforms the BM3D, WNNM, SGWD-HMMs, SSLBD, DnCNN and BUIFD methods by 0.27%, 0.36%, 0.89%, 1.69%, 0.16%, 0.06% respectively.

B. EXPERIMENTS ON BSD IMAGES

1) EXPERIMENTS ON GRAY-SCALE IMAGES

For the grayscale images, 68 grayscale images from the standard Berkeley test dataset BSD68 were used, as shown in Fig. 8, to quantitatively and qualitatively measure the performance of each comparison method. The average *PSNR* and *SSIM* results of different methods on the standard BSD68 dataset under different noise levels (5, 10, 15, 25, 35, 45, 55 and 65) are shown in Table 3 and 4. The highest values of *PSNR* and *SSIM* are highlighted in bold. As shown in Table 3 and 4, the average values of *PSNR* and *SSIM* obtained by GSMRF are highest or one of highest results with different noise levels.

As can be seen from Table 3, our proposed method outperforms the state-of-art denoising methods at different noise levels. Specifically, the average *PSNR* of GSMRF

TABLE 1. ***PSNR*(*dB*) and *SSIM* results of different denoising methods on the synthetic "Squares" image with different noise level.**

Test case	Standard deviation	BM3D	WNNM	SGWD-HMMs	SSLBD	DnCNN	BUIFD	GSMRF
1	$\sigma = 5$	39.76 0.9562	39.62 0.9572	39.87 0.9580	37.68 0.9408	39.94 0.9578	39.82 0.9590	**40.34** **0.9592**
2	$\sigma = 10$	38.86 0.9125	38.69 0.9122	39.02 0.9163	37.01 0.9019	38.86 0.9156	38.82 0.9165	**39.26** **0.9186**
3	$\sigma = 15$	37.75 0.8825	37.78 0.8782	37.78 0.8826	37.46 0.8698	37.82 0.8826	37.80 0.8824	**37.86** **0.8829**
4	$\sigma = 25$	34.26 0.8249	34.55 0.8251	34.28 0.8233	33.01 0.8127	34.56 0.8285	34.50 0.8285	**34.75** **0.8290**
5	$\sigma = 35$	31.49 0.7828	31.50 0.7759	31.55 0.7780	30.21 0.7602	31.62 0.7842	31.60 0.7830	**31.88** **0.7845**
6	$\sigma = 45$	28.32 0.7495	28.26 0.7466	28.01 0.7425	26.34 0.7109	28.60 0.7498	28.58 0.7502	**28.62** **0.7506**
7	$\sigma = 55$	27.10 0.7189	27.06 0.7182	27.05 0.7210	25.78 0.7002	**27.28** **0.7225**	27.20 0.7222	**27.28** **0.7225**
8	$\sigma = 65$	26.02 0.6976	26.05 0.6978	25.88 0.6927	24.79 0.6846	26.08 0.7002	26.12 **0.7006**	**26.15** **0.7006**
	$\bar{\varphi}_{PSNR}$	32.95	32.94	32.93	31.54	33.10	33.06	**33.27**
	$\bar{\varphi}_{SSIM}$	0.8156	0.8139	0.8143	0.7976	0.8177	0.8178	**0.8185**

TABLE 2. ***PSNR*(*dB*) and *SSIM* results of different denoising methods on the synthetic "Circles" image with different noise level.**

Test case	Standard deviation	BM3D	WNNM	SGWD-HMMs	SSLBD	DnCNN	BUIFD	GSMRF
1	$\sigma = 5$	39.65 0.9522	39.42 0.9520	39.30 0.9502	38.76 0.9486	39.64 0.9536	39.85 0.9560	**39.88** **0.9560**
2	$\sigma = 10$	38.46 0.9112	38.28 0.9108	38.22 0.9101	37.72 0.9002	38.45 0.9116	38.62 0.9120	**38.75** **0.9125**
3	$\sigma = 15$	36.25 0.8746	36.29 0.8738	35.87 0.8676	35.27 0.8583	36.22 0.8766	36.38 0.8774	**36.55** **0.8786**
4	$\sigma = 25$	32.36 0.8202	32.25 0.8191	32.15 0.8167	31.33 0.8112	32.56 0.8205	32.65 0.8205	**32.75** **0.8213**
5	$\sigma = 35$	28.79 0.7725	28.58 0.7718	28.07 0.7688	27.20 0.7503	28.82 0.7742	28.90 0.7750	**29.12** **0.7756**
6	$\sigma = 45$	27.02 0.7412	27.06 0.7406	26.68 0.7319	25.96 0.7278	27.24 0.7417	27.28 0.7432	**27.34** **0.7436**
7	$\sigma = 55$	25.92 0.7141	25.96 0.7122	25.20 0.7002	24.88 0.6921	26.06 0.7155	26.08 0.7162	**26.12** **0.7174**
8	$\sigma = 65$	25.05 0.6902	25.08 0.6886	24.56 0.6813	23.67 0.6735	25.16 0.6915	25.22 0.6922	**25.26** **0.6926**
	$\bar{\varphi}_{PSNR}$	31.69	31.62	31.26	30.60	31.77	31.87	**31.97**
	$\bar{\varphi}_{SSIM}$	0.8095	0.8086	0.8034	0.7953	0.8107	0.8116	**0.8122**

outperforms BM3D by the range 0.25 – 0.64, WNNM by the range 0.61 – 0.99, SGWD-HMMs by the range 0.64 – 1.56, SSLBD by the range 1.66 – 3.17, DnCNN by the range 0 – 1.05, BUIFD by the range 0.03 – 1.10.

The BM3D method is often used as a baseline for comparison. According to the conclusions of [37] and [38], few methods can obtain PSNR values that are more than 0.3dB higher on average than BM3D, and the upper limit is 0.7dB. Specifically, the average *PSNR* of our proposed method outperforms BM3D by 0.46dB, 0.37dB, 0.25dB, 0.64dB, 0.4dB, 0.35dB, 0.44dB and 0.60dB at noise levels 5, 10, 15, 25, 35, 45, 55 and 65, respectively. It should be noted that when noise level is 25, our average *PSNR* is 0.64dB higher than BM3D. It is very close to 0.7.

The similar results also appear for the average values of *SSIM* at different noise levels, which are shown in Table 4. Specifically, the average *SSIM* of our method outperforms BM3D by the range 0.25% – 2.70%, WNNM by the range 0.50% – 3.55%, SGWD-HMMs by the range 0.68% – 3.68%, SSLBD by the range 0.97% – 4.42%, DnCNN by the range 0.38% – 1.94%, BUIFD by the range 0% – 1.68%.

FIGURE 8. Standard test dataset BSD68 with 68 grayscale images.

TABLE 3. Average *PSNR*(*dB*) results of different denoising methods on the BSD68 dataset with different noise level.

Dataset	Standard deviation	BM3D	WNNM	SGWD-HMMs	SSLBD	DnCNN	BUIFD	GSMRF
BSD68	$\sigma = 5$	38.12	37.68	37.21	36.92	38.49	38.32	**38.58**
	$\sigma = 10$	34.89	34.27	33.76	33.10	35.01	34.82	**35.26**
	$\sigma = 15$	31.67	31.02	31.28	30.13	31.73	31.75	**31.92**
	$\sigma = 25$	28.71	28.74	27.79	27.25	29.23	29.32	**29.35**
	$\sigma = 35$	27.26	27.05	26.63	25.69	27.55	27.30	**27.66**
	$\sigma = 45$	26.26	25.71	25.28	24.07	26.60	26.58	**26.61**
	$\sigma = 55$	25.31	24.87	24.92	23.46	**25.75**	25.43	**25.75**
	$\sigma = 65$	24.62	24.23	23.78	22.05	24.17	24.12	**25.22**

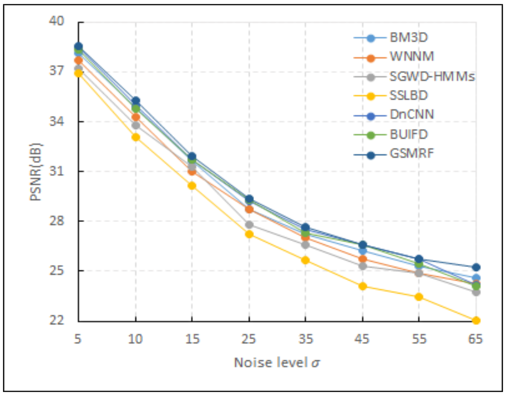


FIGURE 9. Average *PSNRs* (dB) of different methods for BSD68.

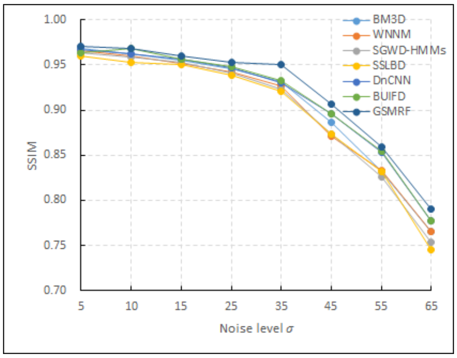


FIGURE 10. Average *SSIMs* of different methods for BSD68.

Figs. 9 and 10 depict the average values of *PSNR* and *SSIM* for BSD68 dataset at the different noise level. It is clear shown in Figs. 9 and 10, our method has improved in varying degrees on average *PSNR* and *SSIM* compared with other state-of-art methods in both low and high noise levels.

TABLE 4. Average *SSIM* results of different denoising methods on the BSD68 dataset with different noise level.

Dataset	Standard deviation	BM3D	WNNM	SGWD-HMMs	SSLBD	DnCNN	BUIFD	GSMRF
BSD68	$\sigma = 5$	0.9687	0.9662	0.9637	0.9598	0.9672	0.9638	**0.9712**
	$\sigma = 10$	0.9620	0.9601	0.9589	0.9527	0.9625	**0.9678**	**0.9678**
	$\sigma = 15$	0.9547	0.9523	0.9532	0.9503	0.9562	0.9568	**0.9600**
	$\sigma = 25$	0.9462	0.9426	0.9410	0.9388	0.9466	0.9476	**0.9532**
	$\sigma = 35$	0.9312	0.9266	0.9228	0.9205	0.9306	0.9332	**0.9500**
	$\sigma = 45$	0.8860	0.8710	0.8729	0.8732	0.8960	0.8960	**0.9065**
	$\sigma = 55$	0.8322	0.8336	0.8256	0.8314	0.8532	0.8540	**0.8592**
	$\sigma = 65$	0.7650	0.7655	0.7530	0.7456	0.7774	0.7775	**0.7898**

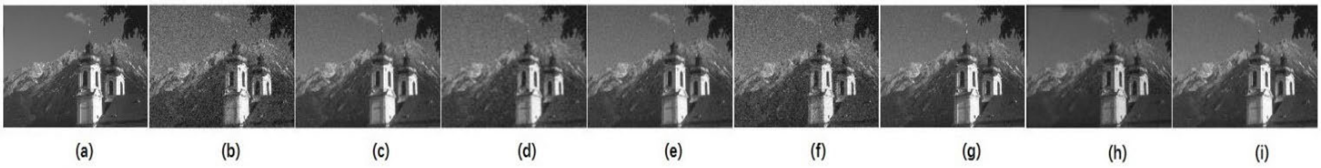


FIGURE 11. Denoising results of one grayscale image from BSD68 with noise level $\sigma = 15$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

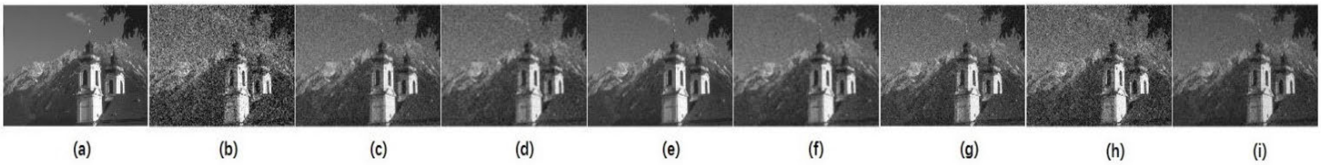


FIGURE 12. Denoising results of one grayscale image from BSD68 with noise level $\sigma = 35$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

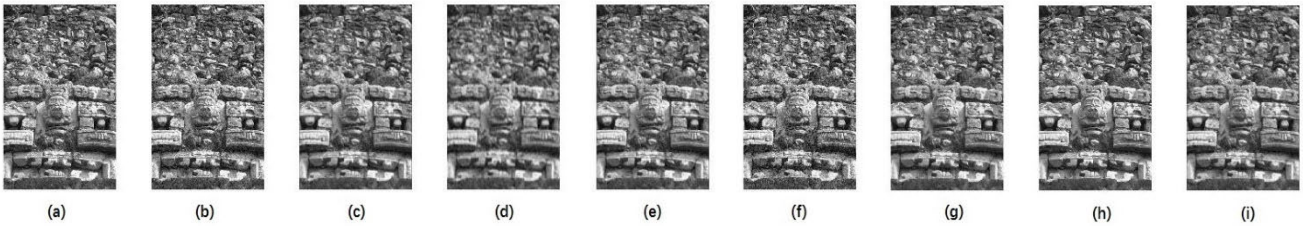


FIGURE 13. Denoising results of one grayscale image from BSD68 with noise level $\sigma = 15$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

Figs. 11-16 show the visual comparison results of three examples selected from BSD68 dataset. The noise levels of each image are 15 and 35, respectively.

For the first example, when the noise level is 15, the denoising results of different methods are shown in Fig. 11. There are obvious noises in the results obtained by WNNM method, while the BM3D, DnCNN, BUIFD and GSMRF methods can remove most of the noises. However, the results obtained by BM3D and DnCNN methods usually yield some artifacts, which cause the denoising images to look blurred. In contrast, BUIFD and GSMRF can yield visually pleasing results, especially our method (if you can zoom in Fig.11 (f) and observe it carefully). When the noise level is 35, as shown in Fig. 12, there are obvious noises and blur in the results obtained by BM3D, WNNM, DnCNN and BUIFD, while only GSMRF obtains visually satisfactory results.

For the second example, when the noise level is 15, the denoising results of different methods are shown in Fig. 13. There are obvious artifacts in the results obtained by WNNM method, which lead the denoising image to look blurred in different degrees, while the result obtained by BM3D has obvious noises. In contrast, DnCNN, BUIFD and GSMRF can yield very clear visual results. When the noise level is 35, as shown in Fig. 14, the denoising results are similar as the noise level equal to 15, which will not be repeated here.

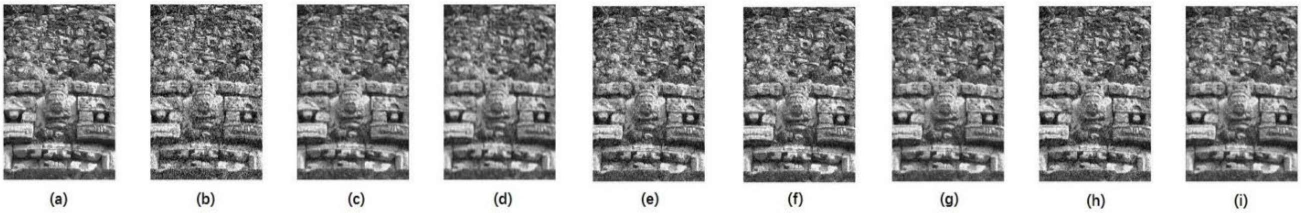


FIGURE 14. Denoising results of one grayscale image from BSD68 with noise level $\sigma = 35$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

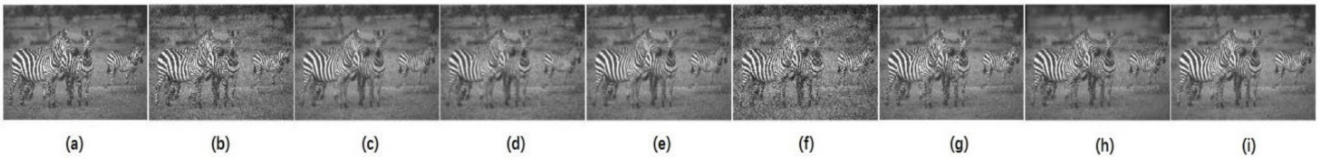


FIGURE 15. Denoising results of one grayscale image from BSD68 with noise level $\sigma = 15$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

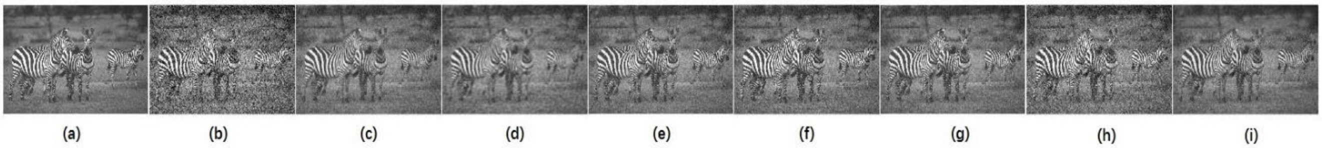


FIGURE 16. Denoising results of one grayscale image from BSD68 with noise level $\sigma = 35$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

TABLE 5. Average *PSNR*(*dB*) results of different denoising methods on the CBSD68 dataset with different noise level.

Dataset	Standard deviation	BM3D	WNNM	SGWD-HMMs	SSLBD	DnCNN	BUIFD	GSMRF
CBSD68	$\sigma = 5$	39.09	38.78	37.87	37.39	39.17	39.13	**39.58**
	$\sigma = 10$	35.99	35.57	34.89	34.01	36.01	35.92	**36.26**
	$\sigma = 15$	33.47	33.22	32.84	31.92	33.36	33.45	**34.02**
	$\sigma = 25$	32.41	32.04	32.12	30.03	32.43	32.32	**32.75**
	$\sigma = 35$	29.06	28.85	27.20	26.87	29.55	29.60	**29.96**
	$\sigma = 45$	26.46	26.31	25.61	24.77	26.51	26.68	**27.03**
	$\sigma = 55$	25.41	25.47	24.42	23.83	25.82	25.85	**25.93**
	$\sigma = 65$	24.62	24.33	23.35	22.56	24.26	24.12	**25.22**

For the third example, when the noise level is 15, the denoising results of different methods are shown in Fig. 15. There are serious noises and artifacts in the results obtained by BM3D and WNNM methods, which lead the denoising images to look blurred in different degrees, while the results obtained by DnCNN and BUIFD have obvious noises. In contrast, GSMRF obtain visually satisfactory results. When the noise level is 35, as shown in Fig. 16, There are obvious artifacts in the results obtained by BM3D and WNNM methods, which lead the denoising image to look blurred in different degrees, while the result obtained by BUIFD have serious noises. In contrast, DnCNN and GSMRF obtain visually satisfactory results.

2) EXPERIMENTS ON COLOR IMAGES

For the color images, 68 color images from the standard Berkeley test dataset CBSD68 were used, as shown in Fig. 17, to quantitatively and qualitatively measure the performance of each comparison method. The average *PSNR* and *SSIM* results of different methods on the standard CBSD68 dataset under different noise levels (5, 10, 15, 25, 35, 45, 55 and 65) are shown in Table 5 and 6. The highest values of *PSNR* and *SSIM* are highlighted in bold.

As shown in Table 5, the average *PSNR* obtained by GSMRF outperforms the state-of-art denoising methods at different noise levels. Specifically, the average *PSNR* of GSMRF outperforms BM3D by the range 0.27 – 0.90, WNNM by the range 0.46 – 1.11, SGWD-HMMs by the range 0.63 – 2.76, SSLBD by the range 2.10 – 3.09, DnCNN by the range 0.11 – 0.96, BUIFD by the range 0.08 – 1.10.

The similar results also appear for *SSIM* values. Specifically, the average *SSIM* of our method outperforms BM3D by the range 0.47% – 5.64%, WNNM by the range 0.65% – 5.55%, SGWD-HMMs by the range 0.57% – 7.85%, SSLBD

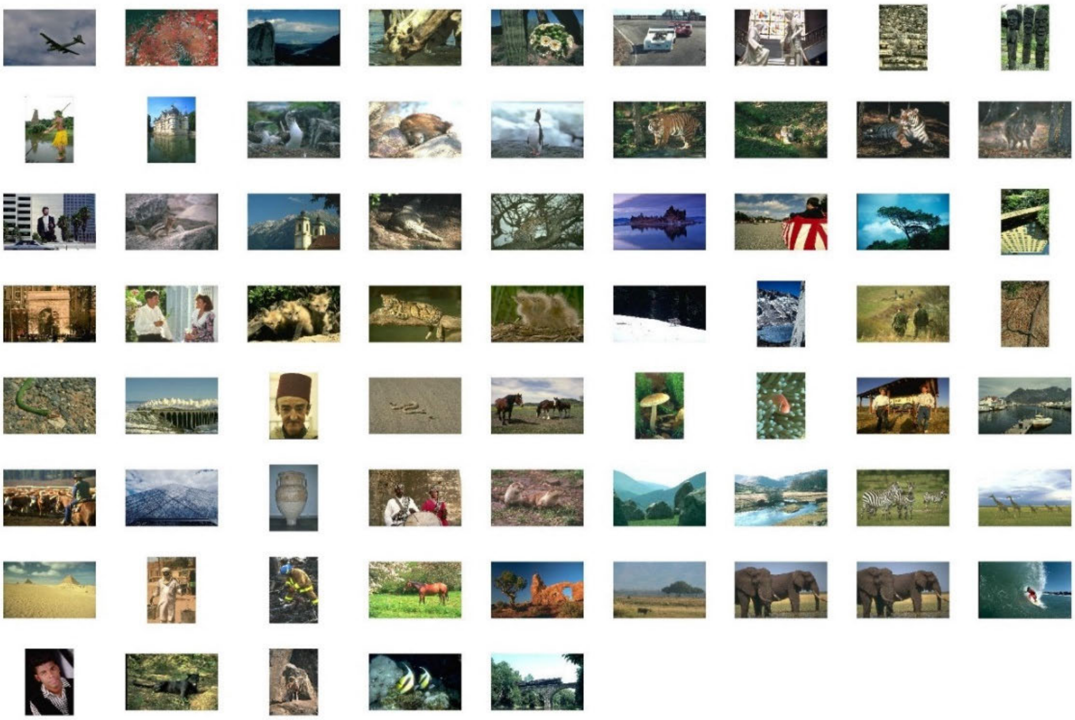


FIGURE 17. Standard test dataset CBSD68 with 68 color images.

TABLE 6. Average *SSIM* results of different denoising methods on the CBSD68 dataset with different noise level.

Dataset	Standard deviation	BM3D	WNNM	SGWD-HMMs	SSLBD	DnCNN	BUIFD	GSMRF
CBSD 68	$\sigma = 5$	0.9737	0.9702	0.9664	0.9589	0.9712	0.9705	**0.9800**
	$\sigma = 10$	0.9511	0.9458	0.9488	0.9375	0.9535	0.9535	**0.9635**
	$\sigma = 15$	0.9342	0.9363	0.9301	0.9221	0.9382	0.9388	**0.9503**
	$\sigma = 25$	0.9190	0.9172	0.9180	0.9068	0.9181	0.9195	**0.9237**
	$\sigma = 35$	0.8638	0.8476	0.8500	0.8421	0.8666	0.8628	**0.9021**
	$\sigma = 45$	0.8062	0.8010	0.7913	0.7835	0.8102	**0.8216**	**0.8216**
	$\sigma = 55$	0.7325	0.7297	0.7139	0.6958	0.7542	0.7561	**0.7582**
	$\sigma = 65$	0.6944	0.6953	0.6723	0.6500	0.7075	0.7095	**0.7508**

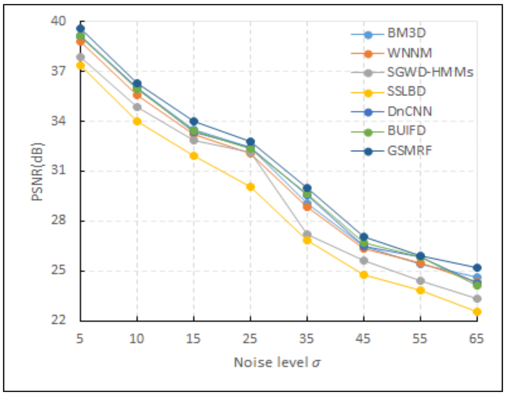


FIGURE 18. Average PSNRs (dB) of different methods for CBSD68.

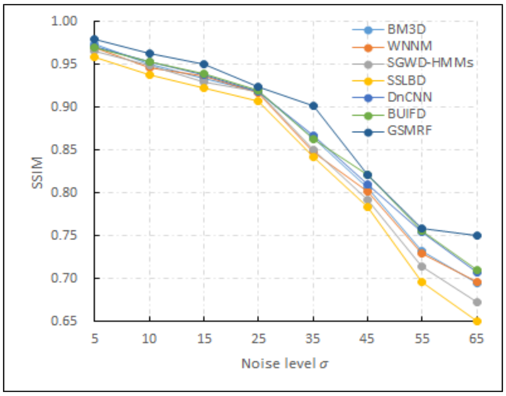


FIGURE 19. FIGURE 19. The average SSIMs of different methods for CBSD68.

by the range 1.69% – 10.08%, DnCNN by the range 0.40% – 4.33%, BUIFD by the range 0% – 4.13%.

Figs. 18 and 19 show the average values of *PSNR* and *SSIM* for CBSD68 dataset at the different noise level. It is clear

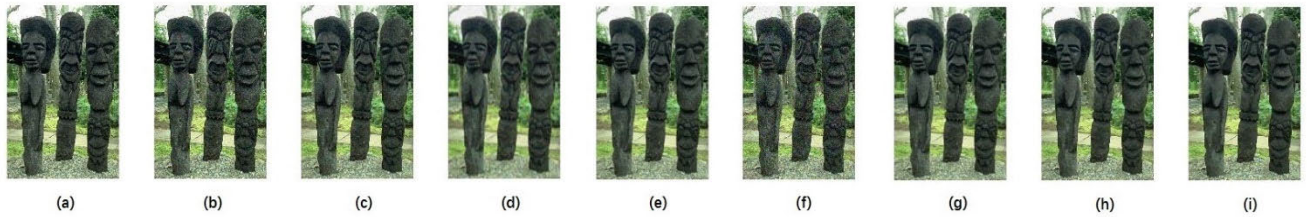


FIGURE 20. Denoising results of one image from CBSD68 color image with noise level $\sigma = 15$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

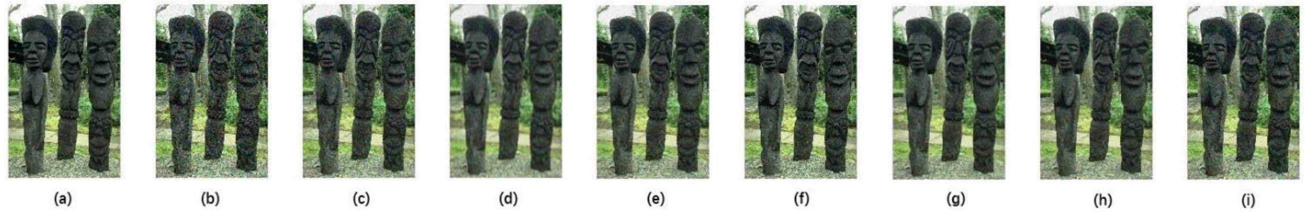


FIGURE 21. Denoising results of one image from CBSD68 color image with noise level $\sigma = 35$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

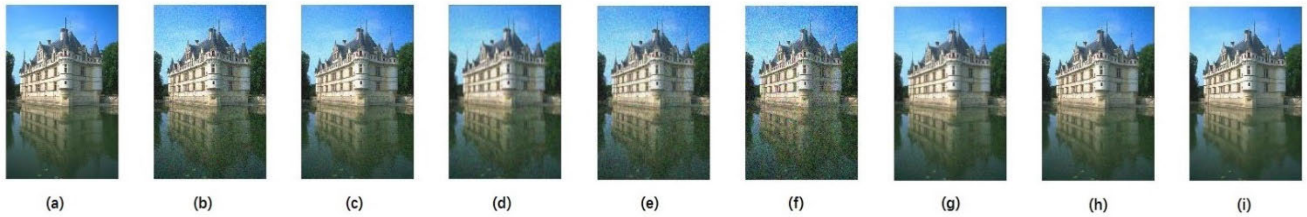


FIGURE 22. Denoising results of one image from CBSD68 color image with noise level $\sigma = 15$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

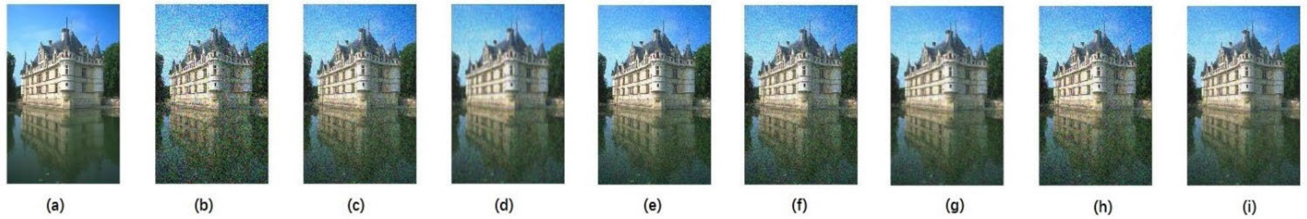


FIGURE 23. Denoising results of one image from CBSD68 color image with noise level $\sigma = 35$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

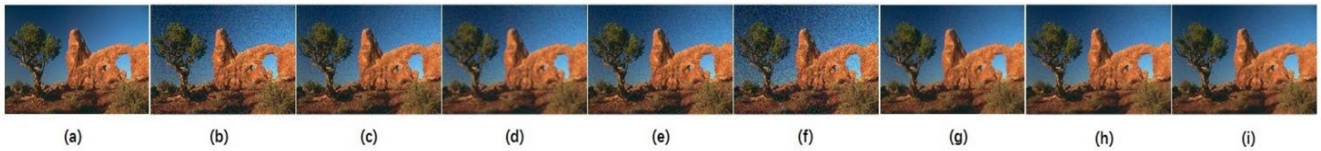


FIGURE 24. Denoising results of one image from CBSD68 color image with noise level $\sigma = 15$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

shown in Figs. 18 and 19, our method has improved in varying degrees on average *PSNR* and *SSIM* compared with other state-of-art methods in both low and high noise levels.

Figs. 20-25 show the visual comparison results of three examples selected from CBSD68 dataset. The noise levels of each image are 15 and 35 respectively.

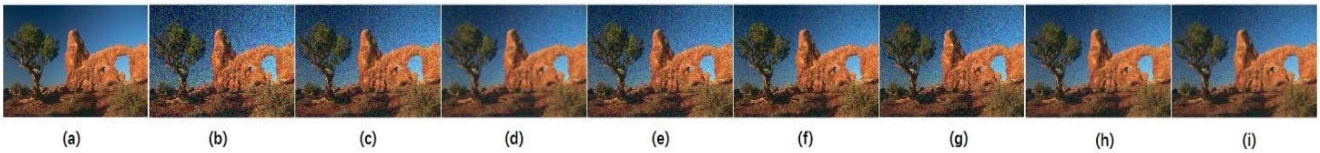


FIGURE 25. Denoising results of one image from CBSD68 color image with noise level $\sigma = 35$. (a) Original image. (b) Noisy image. (c) BM3D. (d) WNNM. (e) SGWD-HMMs. (f) SSLBD. (g) DnCNN. (h) BUIFD. (i) GSMRF.

TABLE 7. Average running times (s) of different denosing methods for the 138 test images with noise level 25.

Method	BM3D	WNNM	SGWD-HMMs	SSLBD	DnCNN	BUIFD	GSMRF
Average Running Time (s)	3.86	828.5	138.79	58.72	4.72	64.79	23.78

For the first example, when the noise level is 15, the denoising results of different methods are shown in Fig. 20. There are obvious noises and artifacts in the results obtained by BM3D, WNNM, DnCNN and BUIFD methods, which lead the denoising images to look blurred in different degrees. In contrast, GSMRF can yield very clear visual results. When the noise level is 35, as shown in Fig. 21, it has serious artifacts in the results obtained by WNNM. The result obtained by BM3D has obvious noises. In contrast, DnCNN, BUIFD and GSMRF obtains visually satisfactory results.

For the second example, when the noise level is 15, the denoising results of different methods are shown in Fig. 22. There are obvious artifacts in the results obtained by BM3D, WNNM and DnCNN methods, which lead the denoising images to look blurred in different degrees. In contrast, BUIFD and GSMRF can yield very clear visual results, especially our method. When the noise level is 35, as shown in Fig. 23, there are obvious artifacts in the results obtained by WNNM and DnCNN, while the results obtained by BM3D and BUIFD have obvious noises. In contrast, only GSMRF obtains visually satisfactory results.

For the third example, when the noise level is 15, the denoising results of different methods are shown in Fig. 24. There are obvious artifacts in the results obtained by WNNM and DnCNN methods, which lead the denoising images to look blurred in different degrees, while the result obtained by BM3D has obvious noises. In contrast, BUIFD and GSMRF obtain visually satisfactory results. When the noise level is 35, as shown in Fig. 25, there are obvious artifacts in the results obtained by WNNM method, which lead the denoising image to look blurred, while the results obtained by BM3D and DnCNN have obvious noises. In contrast, BUIFD and GSMRF obtain visually satisfactory results.

C. CONVERGENCE AND RUNNING TIME

The division of the image is somehow arbitrary. For example, by dividing the whole image into the 4×4 domains as shown in Fig. 2, we obtain 16 different subdomains represented by 16 different hyper-parameters.

We have tested lots of experiments, and setting 16 hyper-parameters was a very satisfactory and effective option. Actually, it is easy to know that more hyper-parameters more time-consuming, less hyper-parameters less accuracy.

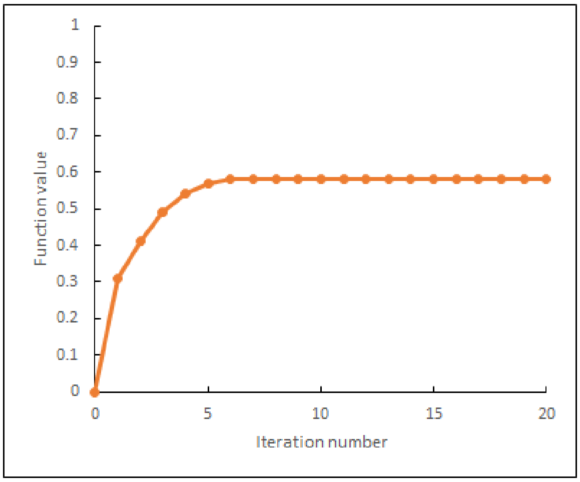


FIGURE 26. Convergence of hyper-parameter estimation.

It should be noted that the optimization of criterion function is an iterative solution process due to the hyper-parameter estimation which uses EM iterative algorithm. The convergence of EM algorithm can be guaranteed theoretically [32], [33].

Fig. 26 shows the convergence of EM algorithm for the hyper-parameter estimation of an image. As can be seen from Fig. 26, the criterion function value can reach convergence after about 5 steps of iteration. The convergence speed is fast.

Beside the qualitative and quantitative assessment, the running time of the image denoising method is another important evaluation. Table 7 reports the average running time obtained by different denoising methods for the above 138 test images (2 synthetic images, 68 grayscale images of BSD68 and 68 color images of CBSD68) with noise level 25. All the experiments are running on the DELL notebook with Intel(R) Core (TM) i5-4300U CPU @ 1.90GHz 2.50GHZ, 16GB memory. The running environment is Matlab (R2015b). As shows in Table 7, the proposed GSMRF method faster than WNNM, SGWD-HMMs, SSLBD and BUIFD methods. And it is inferior to BM3D and DnCNN methods. Although our method is slower than BM3D and DnCNN methods, fortunately, the running time is completely acceptable in many applications. In addition, our GSMRF method is still very competitive by considering its better image quality improvement.

VI. CONCLUSION

Markov random field is actually a discrete random process, and its global characteristics can be described by local characteristics (cluster potential energy). By using a prior probability in terms of the noisy image, combined with the Bayesian criterion, the problem of maximum a posteriori (MAP) estimation can usually be converted into a global optimal value for solving the energy function. This article chooses Gaussian Symmetric Markov random field to describe the local characteristics of the image. That is, because of the Gaussian Symmetric Markov random can fuse the contextual information of the image very well. The Gaussian Symmetric Markov random, whose energy function is quadratic, and the model has multiple parameters that can be well suitable for the model of problem. Another important advantage of our proposed GSMRF filter is that it can theoretically ensure local a posterior probability follows Gaussian distribution. The experiment results indicate that the method proposed in this paper has a good performance to denoising Gaussian noise by comparing with the six state-of-art denoising methods. Besides the above advantages, it should be noted the high time complexity of EM and conjugate gradient algorithms (both are iterative algorithms), which lead to high computing time. In the future, as a development of this work, we will optimize the proposed method here to reduce its time complexity, and extend it to the problem of other noises.

Finally, the purpose of this paper is not a commentary on the best Gaussian denoiser. Conversely, if someone decides to denoising image, then it shows that the proposed method here can denoising as close to the original image as possible.

REFERENCES

[1] J. Varghese, S. Subhash, K. Subramaniam, and K. P. Sridhar, "Adaptive Gaussian notch filter for removing periodic noise from digital images," *IET Image Process.*, vol. 14, no. 8, pp. 1529–1538, Jun. 2020.

[2] A. Awad, "Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise," *Eng. Sci. Technol., Int. J.*, vol. 22, no. 3, pp. 746–753, Jun. 2019.

[3] C. Li, Y. Li, Z. Zhao, L. Yu, and Z. Luo, "A mixed noise removal algorithm based on multi-fidelity modeling with nonsmooth and nonconvex regularization," *Multimedia Tools Appl.*, vol. 78, no. 16, pp. 23117–23140, Aug. 2019.

[4] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 299–314, Feb. 2008.

[5] W. Liu and W. Lin, "Additive white Gaussian noise level estimation in SVD domain for images," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 872–883, Mar. 2013.

[6] J. Nakamura, *Image Sensors and Signal Processing For Digital Still Cameras*. Boca Raton, FL, USA: CRC Press, 2005.

[7] M. Mafi, H. Martin, M. Cabrerizo, J. Andrian, A. Barreto, and M. Adjouadi, "A comprehensive survey on impulse and Gaussian denoising filters for digital images," *Signal Process.*, vol. 157, pp. 236–260, Apr. 2019.

[8] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[9] Z. Zha, X. Liu, X. Huang, H. Shi, Y. Xu, Q. Wang, L. Tang, and X. Zhang, "Analyzing the group sparsity based on the rank minimization methods," 2016, *arXiv:1611.08983*.

[10] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 460–489, 2005.

[11] Y. Weiss and W. T. Freeman, "What makes a good model of natural images?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[12] X. Lan, S. Roth, D. Huttenlocher, and M. J. Black, "Efficient belief propagation with learned higher-order Markov random fields," in *Proc. Eur. Conf. Comput. Vis.*, May 2006, pp. 269–282.

[13] S. Z. Li, *Random Field Modeling in Image Analysis*. London, U.K.: Springer, 2009.

[14] A. Khmag, A. R. Ramli, S. J. bin Hashim, and S. A. R. Al-Haddad, "Additive noise reduction in natural images using second-generation wavelet transform hidden Markov models," *IEEJ Trans. Electr. Electron. Eng.*, vol. 11, no. 3, pp. 339–347, May 2016.

[15] A. Khmag, S. A. R. Al Haddad, R. A. Ramlee, N. Kamarudin, and F. L. Malallah, "Natural image noise removal using nonlocal means and hidden Markov models in transform domain," *Vis. Comput.*, vol. 34, no. 12, pp. 1661–1675, Dec. 2018.

[16] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[17] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[18] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3929–3938.

[19] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[20] M. El Helou and S. Susstrunk, "Blind universal Bayesian image denoising with Gaussian noise level learning," *IEEE Trans. Image Process.*, vol. 29, pp. 4885–4897, 2020.

[21] R. G. Gavaskar and K. N. Chaudhury, "Fast adaptive bilateral filtering," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 779–790, Feb. 2019.

[22] B.-H. Chen, Y.-S. Tseng, and J.-L. Yin, "Gaussian-adaptive bilateral filter," *IEEE Signal Process. Lett.*, vol. 27, pp. 1670–1674, 2020.

[23] B.-H. Chen, H.-Y. Cheng, Y.-S. Tseng, and J.-L. Yin, "Two-pass bilateral smooth filtering for remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[24] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu, "Neighbor2Neighbor: Self-supervised denoising from single noisy images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14776–14785.

[25] J.-L. Yin, B.-H. Chen, and Y. Li, "Highly accurate image reconstruction for multimodal noise suppression using semisupervised learning on big data," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3045–3056, Nov. 2018.

[26] X. Descombes, F. Kruggel, and D. Y. von Cramon, "Spatio-temporal fMRI analysis using Markov random fields," *IEEE Trans. Med. Imag.*, vol. 17, no. 6, pp. 1028–1039, Dec. 1998.

[27] K. Held, E. R. Kops, B. J. Krause, W. M. I. I. I. Wells, R. Kikinis, and H.-W. M´uller-Gartner, "Markov random field segmentation of brain MR images," *IEEE Trans. Med. Imag.*, vol. 16, no. 6, pp. 878–886, Dec. 1997.

[28] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[29] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Trans. Image Process.*, vol. 2, no. 3, pp. 296–310, Jul. 1993.

[30] V. Pascazio and G. Ferraiuolo, "Statistical regularization in linearized microwave imaging through MRF-based MAP estimation: Hyperparameter estimation and image computation," *IEEE Trans. Image Process.*, vol. 12, no. 5, pp. 572–582, May 2003.

[31] S. S. Saquib, C. A. Bouman, and K. Sauer, "ML parameter estimation for Markov random fields with applications to Bayesian tomography," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1029–1044, Jul. 1998.

[32] A. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, Sep. 1977.

[33] J. A. Bilmes, *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*, document ICSI-TR-97-021, International Computer Science Institute, Univ. Berkeley, 1998.

[34] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM J. Numer. Anal.*, vol. 20, no. 3, pp. 626–637, Jun. 1983.

[35] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, Apr. 2009.

[36] H. C. Burger, C. Schuler, and S. Harmeling, "Learning how to combine internal and external denoising methods," in *Proc. Pattern Recognit.*, vol. 2013, pp. 121–130.

[37] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *Proc. CVPR*, Jun. 2011, pp. 2833–2840.

[38] A. Levin, B. Nadler, F. Durand, and W. T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 73–86.

FUSONG XIONG is currently a Senior Lecturer of computer science with the Soochow College, Soochow University, China. His research interests include pattern recognition and image processing.

JIAN ZHANG is currently an Associate Professor of computer science with the Soochow College and the Wenzheng College, Soochow University, China. His research interests include machine learning and design and analysis of algorithm.

ZHIQIANG ZHANG is currently an Associate Professor of computer science with the Soochow College, Soochow University, China. His research interests include AI, pattern recognition, and design and analysis of algorithm.

YUN LING is currently a Professor of computer science with the Soochow College, Soochow University, China. Her research interests include pattern recognition and image processing.