Computers, Materials & Continua
DOI:10.32604/cmc.2021.017300
Article

Tech Science Press

Guided Intra-Patch Smoothing Graph Filtering for Single-Image Denoising

Yibin Tang[1], Ying Chen[2], Aimin Jiang[1], Jian Li[1], Yan Zhou[1,*] and Hon Keung Kwan[3]

[1]College of Internet of Things Engineering, Hohai University, Changzhou, 213022, China
[2]School of Microelectronics and Control Engineering, Changzhou University, Changzhou, 213022, China
[3]Department of Electrical Engineering, University of Windsor, Ontario, N9B 3P4, Canada
[*]Corresponding Author: Yan Zhou. Email: yanzhou@hhu.edu.cn
Received: 26 January 2021; Accepted: 04 March 2021

Abstract: Graph filtering is an important part of graph signal processing and a useful tool for image denoising. Existing graph filtering methods, such as adaptive weighted graph filtering (AWGF), focus on coefficient shrinkage strategies in a graph-frequency domain. However, they seldom consider the image attributes in their graph-filtering procedure. Consequently, the denoising performance of graph filtering is barely comparable with that of other state-of-the-art denoising methods. To fully exploit the image attributes, we propose a guided intra-patch smoothing AWGF (AWGF-GPS) method for single-image denoising. Unlike AWGF, which employs graph topology on patches, AWGF-GPS learns the topology of superpixels by introducing the pixel smoothing attribute of a patch. This operation forces the restored pixels to smoothly evolve in local areas, where both intra- and inter-patch relationships of the image are utilized during patch restoration. Meanwhile, a guided-patch regularizer is incorporated into AWGF-GPS. The guided patch is obtained in advance using a maximum-a-posteriori probability estimator. Because the guided patch is considered as a sketch of a denoised patch, AWGF-GPS can effectively supervise patch restoration during graph filtering to increase the reliability of the denoised patch. Experiments demonstrate that the AWGF-GPS method suitably rebuilds denoising images. It outperforms most state-of-the-art single-image denoising methods and is competitive with certain deep-learning methods. In particular, it has the advantage of managing images with significant noise.

Keywords: Graph filtering; image denoising; MAP estimation; superpixel

1 Introduction

Image denoising aims to remove noise from images, thereby benefitting the subsequent analysis and processing of images and videos. Current image denoising methods are divided into two categories: Model- and deep-learning-based methods. Prior to the success of deep learning methods, model-based methods have been used for decades to exploit the intrinsic attributes of images [1–3]. Most of them resolve the denoising problem using the internal prior of the target noisy image without considering other images. Therefore, they are often considered as single-image

denoising methods. In contrast, deep-learning-based methods employ the external prior from other image databases to recover denoised images. Thus, they can be considered as data-driven methods that require numerous image data [4–6]. Flexible feature learning strategies are adopted in these methods via various deep networks. Because of the feature analysis of external images instead of the target image, deep-learning-based methods have a higher denoising performance than the model-based methods. However, in this study, we focus on the model-based denoising method assuming that only a single noisy image is provided.

In general, model-based denoising methods involve a filtering process, where the input and output data are noisy and denoised images, respectively. Thus, various filtering methods are used in different domains such as spatial, transform, and learned domains. In earlier studies, spatial-domain methods, such as bilateral filtering [7] and non-local means filtering [8], were used for the direct operation on pixels or patches. A spatial smoothing filter was used to remove noise-like components from the target image. However, since this smoothing procedure is only performed in local areas or patch groups, spatial filtering has a limited ability to exploit the statistical information of the entire image. To overcome this problem, transform-domain filters are presented by considering the directional structures in the images. Wavelet and Curvelet filters have been successfully employed for their adequate structural description of basis [9–11]. A popular denoising method, named block matching and three-dimensional (BM3D) filtering, was proposed using a collaborative spatial-wavelet filter [12,13]. It adopts a spatial filtering result to guide sequential wavelet filtering on similar patches. Unfortunately, none of the aforementioned filters are contented-based due to their fixed filter coefficients. Therefore, more advanced models, such as sparse representation and low-rank representation models [3,14–16], have been deployed in different learned domains in recent years. For example, a trilateral weighted sparse coding (TWSC) method is used to estimate data fidelity based on the sparse representation theory [17]. A low-rank approximation approach with adaptive regularizer learning (ARLLR) [18] is presented to shrink the eigenvalues of patches and is highly successful in image denoising. However, it is not yet considered as a filter-based model. The current study proves that the low-rank model is equivalent to a subspace graph filter [19]. The filtering procedure takes place in a graph subspace supported by the eigenvectors of the patch group.

Graph filtering is an essential component of graph signal processing. Its basic idea is to filter the input signal on the network nodes [20]. Once pixels or patches are chosen according to the nodes, graph filtering can adequately fit image denoising. Certain graph polynomial filtering methods are presented to employ various Laplacian matrix regularizers in the existing denoising model [21–23]. Several adaptive graph filtering methods are also proposed, by applying different coefficient shrinkage strategies in the graph-frequency domain. In a pioneer work, an idea lowpass graph filter was designed using a full shrinkage approach in the high graph-frequency band, achieving denoising performance comparable with that of BM3D [24–26]. Given a patch group, the study proves that the eigenvectors of the Laplacian matrix are a set of graph Fourier bases [27]. Recently, an adaptive weighted graph filtering (AWGF) method introduced an effective shrinkage approach in the entire band [19]. Although it theoretically builds a bridge from the existing low-rank model to graph filtering, its denoising performance is barely comparable with that of low-rank denoising methods. In the traditional AWGF method, the graph filter is prioritized, whereas patch attributes are seldom considered for image denoising.

Motivated by the recent progress, we propose a guided intra-patch smoothing AWGF (AWGF-GPS) method for single-image denoising. Our contributions are twofold. (1) Unlike AWGF that uses graph topology on patches, AWGF-GPS learns the superpixel graph to exploit the pixel

smoothing attribute in patches. This operation forces the restored pixels to smoothly evolve in the local area, where both intra- and inter-patch relationships are utilized during patch restoration. (2) A guided-patch regularizer is incorporated into AWGF-GPS. The guided patch is obtained in advance using a maximum-a-posteriori (MAP) probability estimator. By considering the guided patch as a sketch of a denoised patch, AWGF-GPS effectively supervises the patch restoration procedure. Consequently, the reliability of the denoised patch is increased. Experiments demonstrate that our AWGF-GPS method suitably rebuilds denoising images. It outperforms most state-of-the-art model-based methods and is competitive with certain deep-learning methods.

2 Related Work

We briefly review the existing AWGF model [19]. Given a noisy patch group, $\boldsymbol{Y}$, with vectorized patches $\{\boldsymbol{y}_i\}$ in its column, the AWGF model is expressed as

$$\{\tilde{\sigma}_{g,i}\} = \arg\min_{\{\sigma_{g,i}\}} \frac{1}{2\tau} \left\| \boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{U}_g \boldsymbol{\Sigma}_g \boldsymbol{U}_g^T \right\|_F^2 + \sum_{i=1}^{N} g\left(\sigma_{g,i}\right), \tag{1}$$

where τ is a weighted coefficient, $\boldsymbol{U}_g$ is a graph Fourier basis matrix of the graph filter, N is the patch number, and $\boldsymbol{\Sigma}_g$ is a diagonal matrix with the shrinkage coefficients $\{\sigma_{g,i}\}$ as its diagonal entries. Moreover, $g\left(\sigma_{g,i}\right)$ is a prior regularizer, where different regularizers can be adopted to generate various shrinkage strategies on $\sigma_{g,i}$ [28,29]. Sequentially, the denoised patch group, $\boldsymbol{X}$, is obtained as

$$\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{U}_g \tilde{\boldsymbol{\Sigma}}_g \boldsymbol{U}_g^T, \tag{2}$$

where $\tilde{\boldsymbol{\Sigma}}_g$ is a diagonal matrix with the optimal shrinkage coefficients $\{\tilde{\sigma}_{g,i}\}$ as its diagonal entries.

Note that the noisy patch group, $\boldsymbol{Y}$, in Eq. (2) is right-multiplied by the filtering operation of $\boldsymbol{U}_g \tilde{\boldsymbol{\Sigma}}_g \boldsymbol{U}_g^T$. Because the graph filter is built on the graph topology constructed among patch nodes, the AWGF model is an intrinsic patch-based graph filtering model. It focuses on the relationship between patches, where the similarity attribute of the pixels is not considered.

3 Proposed Method

We propose the AWGF-GPS method to employ both the self-similarity of patches and the local similarity of intra-patch pixels. Fig. 1 depicts the filtering flowchart. The superpixel is defined as a pixel set sharing the same location on the patches. A graph is learned using these superpixels to strengthen the smooth attributes of the neighboring pixels. Sequentially, the corresponding graph Fourier bases are obtained from the Laplacian matrix during the graph analysis. The guided patches are evaluated based on their corresponding noisy patches via a MAP estimator. Combined with the graph Fourier bases and the guided patches, the AWGF-GPS model is finally implemented to restore the denoised patches.

3.1 Graph Learning on Superpixels

We obtain the superpixels from the noisy patch group, $\boldsymbol{Y}$, where each row vector of $\boldsymbol{Y}$ represents a superpixel node. Graph learning is then performed on these nodes. Various graph learning methods have been proposed to form graphs with different strategies. However, we prefer

a log-model graph learning approach [30], which builds a connected graph without any isolated nodes. This graph learning model is described as

$$\tilde{\boldsymbol{W}} = \arg\min_{\boldsymbol{W}} \|\boldsymbol{Z} \circ \boldsymbol{W}\|_1 - \alpha \mathbf{1}^T \log \boldsymbol{W}\mathbf{1} + \beta \|\boldsymbol{W}\|_F^2, \tag{3}$$

where $\boldsymbol{W}$ is an adjacent matrix to measure the weighted edges among the superpixel nodes, $\boldsymbol{Z}$ is a pairwise distance matrix for the superpixels, α and β are two weighted coefficients, $\mathbf{1}$ represents a vector of ones, and symbol $\circ$ denotes the Hadamard product.

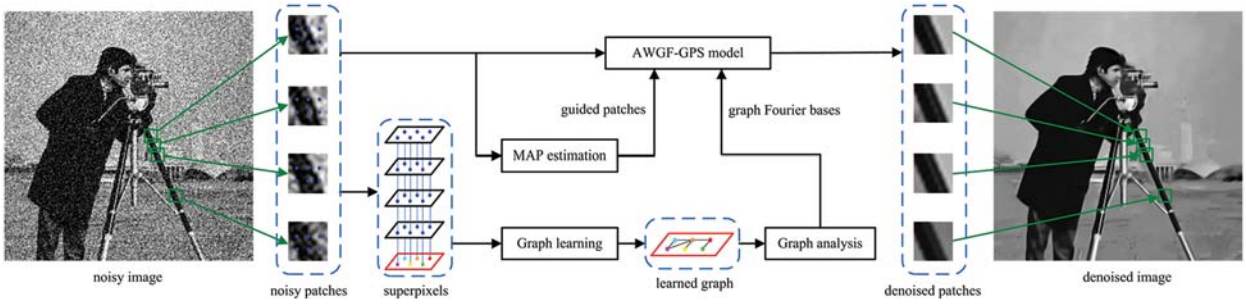

Figure 1: Flowchart of AWGF-GPS denoising

The optimization problem in Eq. (3) can be conveniently solved using the GSP Toolbox [31]. Then, the graph Laplacian matrix is given as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where the diagonal entry of the degree matrix, $\boldsymbol{D}$, is the corresponding row sum of $\boldsymbol{W}$. Finally, the graph Fourier bases, $\boldsymbol{U}_g$, are obtained from the eigenvectors of $\boldsymbol{L}$.

3.2 Guided Patch Estimation

We employ the guided patches as a regularizer in our AWGF-GPS model. Here, a MAP estimator is used to generate the guided patch group, $\hat{\boldsymbol{X}}$, that is viewed as a sketch of the denoised patch group, $\boldsymbol{X}$. For convenience, we interpret the MAP estimator at the pixel level [32] and then extend it to the patch operation level. Given a noisy pixel, $y = x + n$, with clean pixel x and noise n, where n follows a Gaussian distribution, $\mathcal{N}(0, \sigma_n^2)$, with a zero mean value and standard deviation σ_n, the conditional probability of x can be expressed as

$$p(x \mid y) = \frac{1}{\sqrt{2\pi\hat{\sigma}_x^2}} \exp\left\{-\frac{(x-\hat{\mu})^2}{2\hat{\sigma}_x^2}\right\}. \tag{4}$$

In Eq. (4), symbols $\hat{\mu}$ and $\hat{\sigma}_x^2$ are respectively defined as

$$\hat{\mu} = \frac{\sigma_n^2 \bar{x} + \sigma_x^2 y}{\sigma_n^2 + \sigma_x^2}, \tag{5}$$

$$\hat{\sigma}_x^2 = \frac{\sigma_n^2 \sigma_x^2}{\sigma_n^2 + \sigma_x^2}, \tag{6}$$

where $\bar{x}$ and σ_x are the mean value and standard deviation of x, respectively. Hence, we define the guided pixel, $\hat{x}$, as the conditional expected value of x with

$$\begin{aligned}\hat{x} &= \int_{-\infty}^{\infty} x \cdot p\left(x \mid y\right) dx \\ &= \frac{\bar{x}}{1+S} + \frac{y}{1+1/S}\end{aligned}, \tag{7}$$

where $S = \sigma_x^2/\sigma_n^2$ is the signal-to-noise ratio (SNR). Inspired by the pixel estimation of $\hat{x}$, the guided group, $\hat{\boldsymbol{X}}$, can be written as

$$\hat{\boldsymbol{X}} = \frac{\bar{\boldsymbol{X}}}{1+S} + \frac{\boldsymbol{Y}}{1+1/S}, \tag{8}$$

where $\bar{\boldsymbol{X}}$ is the mean patch group composed of the corresponding guided pixels. In Eq. (8), we assume that each pixel in the patch group shares the same SNR.

Unfortunately, the guided group, $\hat{\boldsymbol{X}}$, in Eq. (8) is impractical because there is no exquisite formulation for $\bar{\boldsymbol{X}}$ and S. As a result, we generate these two parameters by an alternative approach. To estimate the SNR, we approximate S as

$$S \approx \frac{\mathrm{var}\left(\boldsymbol{Y}\right) - \sigma_n^2}{\sigma_n^2}, \tag{9}$$

where $\mathrm{var}\left(\boldsymbol{Y}\right)$ is the mean deviation of pixel for patch group $\boldsymbol{Y}$.

We estimate the mean patch group, $\bar{\boldsymbol{X}}$, using an ideal lowpass graph filter as

$$\bar{\boldsymbol{X}} \approx \boldsymbol{U}_g \tilde{\boldsymbol{\Sigma}}_{g,M} \boldsymbol{U}_g^T \boldsymbol{Y}, \tag{10}$$

where the diagonal matrix, $\tilde{\boldsymbol{\Sigma}}_{g,M}$, contains only the first M nonzero diagonal entries. These entries are described as

$$\sigma_{g,i} = \begin{cases} 1 & i \le M \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

The optimal value of M is further achieved under the noise control of σ_n^2 as

$$\tilde{M} = \arg\min_{M} \left| \mathrm{var}\left(\boldsymbol{U}_g \boldsymbol{\Sigma}_{g,M} \boldsymbol{U}_g^T \boldsymbol{Y}\right) + \sigma_n^2 - \mathrm{var}\left(\boldsymbol{Y}\right) \right|. \tag{12}$$

3.3 AWGF-GPS Model

Using graph Fourier bases and guided patches, we define the AWGF-GPS model as

$$\left\{\tilde{\sigma}_{g,i}\right\} = \arg\min_{\{\sigma_{g,i}\}} \frac{1}{2\tau} \left\| \boldsymbol{Y} - \boldsymbol{U}_g \boldsymbol{\Sigma}_g \boldsymbol{U}_g{}^T \boldsymbol{Y} \right\|_F^2 + \sum_{i=1}^{N'} g\left(\sigma_{g,i}\right) + \gamma \left\| \hat{\boldsymbol{X}} - \boldsymbol{U}_g \boldsymbol{\Sigma}_g \boldsymbol{U}_g{}^T \boldsymbol{Y} \right\|_F^2, \tag{13}$$

where γ is a weighted coefficient and N' is the superpixel number. Then the denoised patch group, $\boldsymbol{X}$, is obtained as

$$\boldsymbol{X} = \boldsymbol{U}_g \tilde{\boldsymbol{\Sigma}}_g \boldsymbol{U}_g^T \boldsymbol{Y}. \tag{14}$$

We should mention some aspects of the filtering procedure from Eqs. (13) to (14). Unlike the definition in Eq. (2), the noisy patch group, $\boldsymbol{Y}$, in Eq. (14) is left-multiplied by the filtering operation of $\boldsymbol{U}_g\tilde{\boldsymbol{\Sigma}}_g\boldsymbol{U}_g^T$. In this case, the row combination process is employed for $\boldsymbol{Y}$. Since each row vector of $\boldsymbol{Y}$ represents a superpixel, it conducts a superpixel-based filtering model. Moreover, we aim for the denoising patches described in Eq. (13) to be centralized with $\hat{\boldsymbol{X}}$, because the guided patch group, $\hat{\boldsymbol{X}}$, is another denoised result for clean patches. These guide patches provide an optimal direction for denoising.

To simplify Eq. (13), we transform the model by merging its first and third terms as

$$\left\{\tilde{\sigma}_{g,i}\right\} = \arg\min_{\{\sigma_{g,i}\}} (1+2\lambda\tau) \left\| \boldsymbol{U}_g\boldsymbol{\Sigma}_g{\boldsymbol{U}_g}^T\boldsymbol{Y} - \frac{1}{1+2\lambda\tau}\left(\boldsymbol{Y}+2\lambda\tau\hat{\boldsymbol{X}}\right)\right\|_F^2 + 2\tau\sum_{i=1}^{N'} g\left(\sigma_{g,i}\right). \tag{15}$$

Subsequently, we cast Eq. (15) in the trace form as

$$\left\{\tilde{\sigma}_{g,i}\right\} = \arg\min_{\{\sigma_{g,i}\}} (1+2\lambda\tau)\, trace\left(\boldsymbol{\Sigma}_g^2{\boldsymbol{U}_g}^T\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{U}_g - 2\boldsymbol{\Sigma}_g{\boldsymbol{U}_g}^T\boldsymbol{Y}\boldsymbol{E}^T\boldsymbol{U}_g\right) + 2\lambda\tau\sum_{i=1}^{N'} g\left(\sigma_{g,i}\right), \tag{16}$$

where $\boldsymbol{E} = (1+2\lambda\tau)^{-1}\left(\boldsymbol{Y}+2\lambda\tau\hat{\boldsymbol{X}}\right)$.

Then, we provide the setting of the prior regularizer, $g\left(\sigma_{g,i}\right)$. According to the AWGF approach [19], regularizer $g\left(\sigma_{g,i}\right)$ is expressed as

$$g\left(\sigma_{g,i}\right) = \log\left|\eta_i\sigma_{g,i}+\varepsilon\right|, \tag{17}$$

where ε is a small positive constant and η_i is a weighted coefficient. By defining the projection matrix, $\boldsymbol{P} = \boldsymbol{Y}^T\boldsymbol{U}_g$, with its vectors $\left\{\boldsymbol{p}_i\right\}$, we set η_i as the projected energy with $\eta_i^2 = \boldsymbol{p}_i^T\boldsymbol{p}_i$.

Because the shrinkage coefficient matrix, $\boldsymbol{\Sigma}_g$, is diagonal, the optimization problem in Eq. (16) can be decomposed into a set of subproblems by the given $g\left(\sigma_{g,i}\right)$ of Eq. (17). For each shrinkage coefficient, $\sigma_{g,i}$, the corresponding subproblem is written as

$$\tilde{\sigma}_{g,i} = \arg\min_{\sigma_{g,i}} (1+2\lambda\tau)\left(\eta_i\sigma_{g,i} - \frac{f_i}{\eta_i}\right)^2 + 2\lambda\tau\log\left|\eta_i\sigma_{g,i}+\varepsilon\right|, \tag{18}$$

where f_i is the i-th diagonal entry of ${\boldsymbol{U}_g}^T\boldsymbol{Y}\boldsymbol{E}^T\boldsymbol{U}_g$. Considering the shrinkage coefficient solution in AWGF as a reference, Eq. (18) also provides a closed-form solution as

$$\tilde{\sigma}_{g,i} = \frac{1}{\eta_i}\max\left(\frac{f_i}{\eta_i} - \varepsilon - \frac{2\left(\frac{1+2\lambda\tau}{\lambda\tau} - \varepsilon\frac{f_i}{\eta_i}\right)}{\sqrt{\left(\frac{f_i}{\eta_i}-\varepsilon\right)^2 - 4\left(\frac{1+2\lambda\tau}{\lambda\tau} - \varepsilon\frac{f_i}{\eta_i}\right)}}, 0\right). \tag{19}$$

Thus, the AWGF-GPS method completes its coefficient shrinkage procedure.

3.4 AWGF-GPS Denoising Framework

We present the iterative AWGF-GPS denoising framework in Algorithm 1. Given an intermediate noisy image, $\mathbf{Y}^{(k)}$, in the k-th iteration, we deal with its j-th noisy patch group, $\boldsymbol{Y}_j$, in four steps.

Algorithm 1: Image denoising by AWGF-GPS

Input: Noisy image $\mathbf{Y}$, noise standard variance σ_n.
1: Initialize intermediate noisy and denoised images with $\mathbf{Y}^{(0)} = \mathbf{X}^{(0)} = \mathbf{Y}$.
2: **For** $k = 1 : K$ **do**
3: Set intermediate noisy image $\mathbf{Y}^{(k)} = \alpha \mathbf{X}^{(k-1)} + (1-\alpha)\,\mathbf{Y}$.
4: **For** each patch $\boldsymbol{y}_j$ of $\mathbf{Y}^{(k)}$ **do**
5: Find similar patches of $\boldsymbol{y}_j$ to form patch group $\boldsymbol{Y}_j$.
6: Learn the graph from superpixels using Eq. (3).
7: Calculate eigenvectors $\boldsymbol{U}_g$ from adjacent matrix $\boldsymbol{W}$.
8: Define noise standard variance $\sigma_{n,j}$ using Eq. (20).
9: Achieve guided patches $\hat{\boldsymbol{X}}_j$ using Eq. (8) with its SNR S_j and mean patch group $\bar{\boldsymbol{X}}_j$.
10: Obtain shrinkage coefficients $\{\tilde{\sigma}_{g,i}\}$ using Eq. (19).
11: Obtain restored patch group $\boldsymbol{X}_j$ using Eq. (14).
12: **end for**
13: Aggregate all $\{\boldsymbol{X}_j\}$ to form intermediate denoised image $\mathbf{X}^{(k)}$.
14: **end for**
Output: Optimal denoised image $\tilde{\mathbf{X}} = \mathbf{X}^{(K)}$.

First, we obtain the Laplacian matrix, $\boldsymbol{L}$, from the learned superpixel-based graph, $\boldsymbol{W}$, in Eq. (3) and obtain its graph Fourier bases, $\boldsymbol{U}_g$. Second, the j-th guided patch group, $\hat{\boldsymbol{X}}_j$, is estimated using Eq. (8), where the corresponding SNR, S_j, and the mean patch group, $\bar{\boldsymbol{X}}_j$, are defined by Eqs. (9) and (10), respectively. Here, the noise standard variance, $\sigma_{n,j}$, of $\boldsymbol{Y}_j$ is given by

$$\sigma_{n,j} = \sqrt{\max\left(\sigma_n^2 - \mathrm{var}\left(\boldsymbol{Y}_j - \boldsymbol{Y}_j'\right), 0\right)}, \tag{20}$$

where $\boldsymbol{Y}_j'$ is the patch group of noisy image $\mathbf{Y}$ sharing the same patch index of $\boldsymbol{Y}_j$. Third, the AWGF-GPS model is applied to the patch group, $\boldsymbol{Y}_j$, to achieve its shrinkage coefficients $\{\tilde{\sigma}_{g,i}\}$ using Eq. (19) and the intermediate denoised patches, $\boldsymbol{X}_j$, using Eq. (14). Finally, we assemble all denoised patches $\{\boldsymbol{X}_j\}$ into the k-th iterative denoised image, $\mathbf{X}^{(k)}$. The optimal denoised image, $\tilde{\mathbf{X}}$, is obtained after the K-th iteration.

4 Experimental Results

We compare the AWGF-GPS method with several state-of-the-art model-based denoising methods, including BM3D [12], TWSC [17], ARLLR [18], and AWGF [19]. A deep-learning image denoising method, named the fast and flexible denoising convolutional neural network (FFDNET) [5], is adopted to show the gap between the model-and deep-learning-based denoising methods. Moreover, because our method is derived from AWGF, the denoising parameters are also inherited from those in AWGF. However, for the weighted coefficients in Eq. (13), parameters

τ and γ are set as $\tau = \sqrt{N'}\sigma_{n,j}^2$ and $\gamma = 0.1\tau^{-1}$, respectively. A few clean images are shown in Fig. 2 as test materials. The noisy images are generated by adding Gaussian noise.

Figure 2: Clean images. From left to right, the images on the top line are named C. Man, House, Peppers, Starfish, Man, Monarch, and Lena, and those on the bottom line are named Airplane, Boats, Parrot, Barbara, Couple, Montage, and Hills

A denoised image comparison of the noise standard deviation of $\sigma_n = 40$ is shown in Fig. 3. As a benchmark, BM3D achieves an acceptable denoising performance through collaborative spatial-wavelet filtering. TWSC is better than BM3D with its efficiently learned features of noisy images via sparse representation. ARLLR outperforms the other methods. This indicates that the eigenvalue shrinkage approach is powerful for managing noise. The noise is sufficiently removed in the high "rank-bands" corresponding to the small eigenvalues of the patch group. AWGF emphasizes graph filtering but has less consideration for the image attributes. This leads to a denoising performance that is barely comparable with that of ARLLR. Our method is proven to be the best. The denoised images derived from AWG-GPS are smoothed with a few artificial textures by employing the intra-patch smoothing attribute among the superpixels and the supervision from guided patches. However, FFDNET achieves an outstanding performance in restoring images more naturally. Unlike model-based methods, it benefits from learning the prior from the external images.

The denoised image comparison of the noise standard deviation of $\sigma_n = 70$ is shown in Fig. 4. BM3D obtains good results. TWSC and ARLLR are better than BM3D due to their self-learned features from noisy images. We observe that ARLLR suffers from stronger artificial textures (House). ARLLR is an eigenvalue shrinkage method that uses the statistical information of patch groups via singular value decomposition (SVD). However, it seldom focuses on the pixel smoothing attribute in local areas. AWGF outperforms the aforementioned methods. This proves that graph filtering on the learned graph is useful to alleviate the artificial textures, since the restored patches are required to smoothly evolve on the graph. The proposed AWGF-GPS further smoothens the denoised images, which is owing to the superpixel-based graph, where the similarities between the patches and intra-patch pixels are fully exploited. The shrinkage approach is also efficiently conducted using guided patches. Note that the denoising performance of FFDNET is unsatisfactory. For example, the eaves area of the house is not sufficiently restored. This phenomenon is caused by feature mismatch. In this case, the external prior from the image

databases cannot precisely guide the patch reconstruction for denoised images because the features of clean images are significantly contaminated by noise and are difficult to recognize.

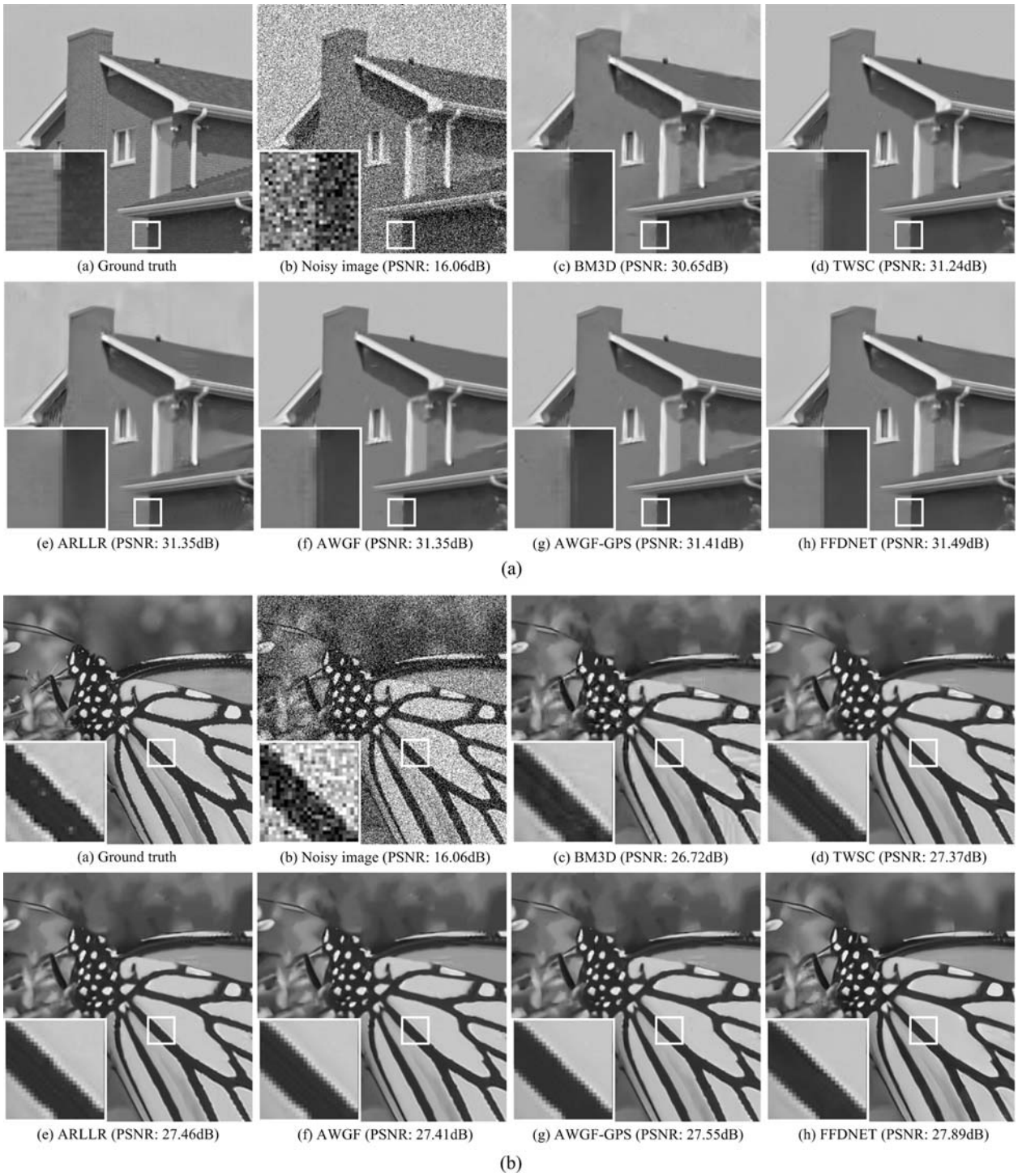

Figure 3: Denoised image comparison of the noise deviation of $\sigma_n = 40$. (a) House (b) Monarch

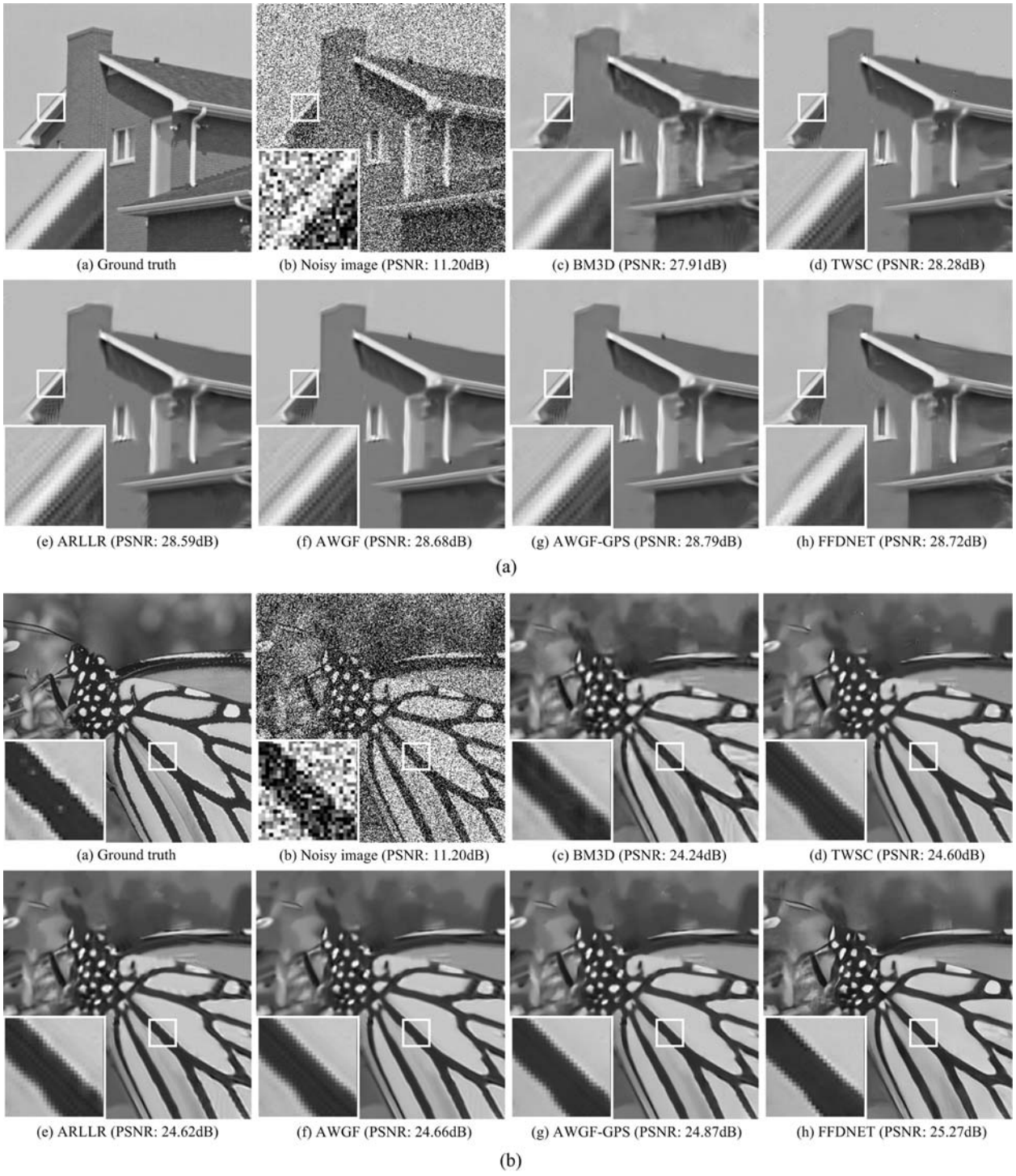

Figure 4: Denoised image comparison of the noise deviation of $\sigma_n = 70$. (a) House (b) Monarch

Table 1: PSNR (dB) results of different denoising methods

	C. Man	*House*	*Pepper*	*Starfish*	*Man*	*Monarch*	*Lena*	*Airplane*	*Boats*	*Parrot*	*Barbara*	*Couple*	*Montage*	*Hills*	*Average*
$\sigma_n = 20$															
BM3D	30.49	33.77	31.29	29.67	30.59	30.35	33.05	29.55	30.88	29.96	31.78	30.76	33.34	30.72	31.16
TWSC	30.57	33.89	31.43	30.13	30.72	30.88	33.19	29.66	30.97	30.11	32.32	30.82	33.44	30.77	31.35
ARLLR	30.74	34.03	31.55	30.29	30.74	31.11	33.12	29.90	30.98	30.20	32.20	30.80	33.66	30.78	31.44
AWGF-GPS	30.80	34.05	31.65	30.31	30.81	31.13	33.12	30.00	30.96	30.23	32.13	30.84	33.72	30.82	31.47
FFDNET	31.13	34.15	32.00	30.51	31.10	31.22	33.51	30.16	31.22	30.44	31.21	31.24	33.60	30.95	31.60
$\sigma_n = 30$															
BM3D	28.64	32.09	29.28	27.65	28.86	28.36	31.26	27.56	29.12	28.12	29.81	28.87	31.21	29.16	29.28
TWSC	28.69	32.40	29.41	28.03	28.92	28.78	31.41	27.67	29.19	28.27	30.35	28.93	31.31	29.18	29.47
ARLLR	28.78	32.55	29.49	28.07	28.98	28.91	31.43	27.82	29.23	28.32	30.31	28.97	31.44	29.23	29.54
AWGF-GPS	28.86	32.63	29.61	28.14	29.03	29.00	31.44	27.93	29.25	28.38	30.24	28.99	31.57	29.25	29.59
FFDNET	29.33	32.68	30.08	28.40	29.34	29.24	31.83	28.21	29.52	28.67	29.24	29.44	31.65	29.43	29.79
$\sigma_n = 40$															
BM3D	27.18	30.65	27.70	26.06	27.65	26.72	29.86	26.08	27.74	26.69	27.99	27.48	29.36	27.99	27.80
TWSC	27.42	31.24	27.97	26.55	27.72	27.37	30.11	26.35	27.89	27.05	28.82	27.58	29.69	28.05	28.13
ARLLR	27.47	31.35	28.06	26.59	27.78	27.46	30.11	26.49	27.94	27.09	28.75	27.61	29.78	28.11	28.18
AWGF-GPS	27.56	31.41	28.14	26.64	27.83	27.55	30.13	26.58	27.96	27.15	28.68	27.63	29.92	28.13	28.24
FFDNET	28.06	31.49	28.65	26.91	28.19	27.89	30.62	26.90	28.31	27.50	27.83	28.14	30.20	28.39	28.51
$\sigma_n = 50$															
BM3D	26.13	29.69	26.68	25.04	26.81	25.82	29.05	25.01	26.78	25.90	27.23	26.46	27.86	27.19	26.83
TWSC	26.46	30.17	26.88	25.41	26.81	26.27	29.08	25.38	26.88	26.11	27.54	26.48	28.35	27.19	27.07
ARLLR	26.42	30.33	26.91	25.43	26.94	26.32	29.25	25.42	26.97	26.09	27.79	26.64	28.15	27.33	27.14
AWGF-GPS	26.53	30.45	27.06	25.51	27.00	26.45	29.28	25.55	27.00	26.19	27.69	26.68	28.40	27.36	27.23
FFDNET	27.12	30.45	27.54	25.76	27.32	26.85	29.67	25.94	27.37	26.60	26.71	27.14	29.02	27.60	27.51
$\sigma_n = 70$															
BM3D	24.62	27.91	25.07	23.56	25.56	24.24	27.57	23.75	24.40	24.49	25.47	25.00	25.85	25.93	25.24
TWSC	24.94	28.28	25.23	23.72	25.48	24.60	27.44	23.94	25.31	24.69	25.64	24.79	26.33	25.81	25.44
ARLLR	24.86	28.59	25.25	23.78	25.68	24.62	27.85	24.00	25.58	24.64	26.17	25.18	26.03	26.14	25.60
AWGF-GPS	25.01	28.84	25.51	23.92	25.73	24.86	27.90	24.17	25.63	24.80	26.10	25.22	26.30	26.18	25.73
FFDNET	25.68	28.73	25.90	24.13	26.02	25.27	28.24	24.53	25.96	25.23	24.99	25.63	27.15	26.40	25.99
$\sigma_n = 100$															
BM3D	23.08	25.87	23.59	22.10	24.22	22.52	25.95	22.11	23.97	22.96	23.62	23.51	23.89	24.58	23.71
TWSC	23.22	25.87	23.24	21.98	24.03	22.66	25.74	22.22	23.67	23.05	23.63	23.07	24.14	24.29	23.63
ARLLR	23.35	26.66	23.45	22.23	24.36	22.95	26.21	22.55	24.11	23.19	24.37	23.56	24.12	24.76	23.99
AWGF-GPS	23.50	26.79	23.64	22.30	24.43	23.14	26.27	22.71	24.16	23.30	24.31	23.59	24.27	24.80	24.09
FFDNET	22.61	24.21	22.47	21.30	22.93	22.01	24.09	21.82	22.86	22.41	21.90	22.52	23.24	23.22	22.67

Notes: *For each test image, the corresponding denoised images with the best PSNR are marked in blue, and those with the second highest PSNR values are marked in red.

The statistical results of the PSNR and SSIM are listed in Tabs. 1 and 2. We use BM3D as a baseline, for it achieves an acceptable performance for all the images and noise levels. TWSC outperforms BM3D because it is a learned domain method. In the sparse representation model, dictionary learning aims to catch patch features as atoms, whereas sparse coding focuses on patch restoration. However, as the noise level increases, the performance of the TWSC degenerates significantly. In this case, the atoms are distorted by direct feature learning from the noisy image. ARLLR is better than the former methods. The shrinkage approach on the patch group is effective in dealing with noise because the relationship among patches is exploited. Our AWGF-GPS method performs the best among these model-based denoising methods. As previously indicated,

its coefficient shrinkage strategy is presented on the superpixel graph, where both attributes, within and among the patches, are considered. The guided patch regularizer further enhances the denoising performance. We also note that our method may suffer from an over-smoothing problem. It is inappropriate to deal with images (e.g., Barbara) that contain strong textures. The AWGF-GPS model attempts to design a lowpass graph filter. Because the components of the texture are centralized in high graph-frequency bands, the AWGF-GPS filter is unsatisfactory. FFDNET is best in terms of the noise standard deviation of $\sigma_n \leq 70$ but the worst for $\sigma_n = 100$. The use of an external prior is effective in improving denoising performance. However, once the feature mismatch phenomenon occurs, the denoised result significantly deteriorates and becomes inferior to the results of the model-based methods. Compared with FFDNET, our method has the advantage in coping with large noise.

Table 2: SSIM results of different denoising methods

	C. Man	*House*	*Pepper*	*Starfish*	*Man*	*Monarch*	*Lena*	*Airplane*	*Boats*	*Parrot*	*Barbara*	*Couple*	*Montage*	*Hills*	*Average*
$\sigma_n = 20$															
BM3D	0.8755	0.8726	0.8868	0.8748	0.8333	0.9179	0.8772	0.8762	0.8259	0.8696	0.9054	0.8476	0.9400	0.8040	0.8719
TWSC	0.8789	0.8716	0.8899	0.8805	0.8369	0.9249	0.8817	0.8803	0.8264	0.8726	0.9139	0.8464	0.9447	0.8044	0.8752
ARLLR	0.8785	0.8716	0.8910	0.8822	0.8350	0.9245	0.8787	0.8815	0.8250	0.8719	0.9107	0.8437	0.9432	0.8031	0.8743
AWGF-GPS	0.8799	0.8703	0.8913	0.8819	0.8382	0.9243	0.8782	0.8836	0.8264	0.8733	0.9093	0.8440	0.9419	0.8058	0.8749
FFDNET	0.8939	0.8731	0.9040	0.8938	0.8549	0.9336	0.8902	0.8898	0.8412	0.8871	0.9041	0.8644	0.9484	0.8214	0.8857
$\sigma_n = 30$															
BM3D	0.8373	0.8480	0.8505	0.8289	0.7802	0.8822	0.8449	0.8372	0.7795	0.8318	0.8687	0.7947	0.9113	0.7504	0.8318
TWSC	0.8375	0.8507	0.8558	0.8352	0.7797	0.8935	0.8510	0.8436	0.7778	0.8347	0.8835	0.7934	0.9207	0.7465	0.8360
ARLLR	0.8399	0.8523	0.8567	0.8357	0.7818	0.8924	0.8502	0.8438	0.7792	0.8346	0.8811	0.7945	0.9187	0.7498	0.8365
AWGF-GPS	0.8420	0.8529	0.8582	0.8382	0.7849	0.8939	0.8506	0.8457	0.7802	0.8344	0.8794	0.7936	0.9191	0.7498	0.8374
FFDNET	0.8594	0.8541	0.8734	0.8506	0.8017	0.9048	0.8627	0.8572	0.7959	0.8520	0.8657	0.8189	0.9279	0.7658	0.8493
$\sigma_n = 40$															
BM3D	0.8057	0.8256	0.8158	0.7828	0.7374	0.8446	0.8152	0.7985	0.7387	0.7992	0.8225	0.7469	0.8806	0.7069	0.7943
TWSC	0.8059	0.8360	0.8259	0.7949	0.7366	0.8628	0.8247	0.8116	0.7367	0.8056	0.8489	0.7465	0.8975	0.7018	0.8025
ARLLR	0.8058	0.8348	0.8263	0.7953	0.7387	0.8597	0.8220	0.8106	0.7394	0.8043	0.8445	0.7475	0.8929	0.7064	0.8020
AWGF-GPS	0.8094	0.8371	0.8290	0.7975	0.7421	0.8614	0.8236	0.8127	0.7399	0.8083	0.8427	0.7466	0.8938	0.7070	0.8037
FFDNET	0.8302	0.8391	0.8440	0.8125	0.7603	0.8773	0.8397	0.8292	0.7582	0.8253	0.8275	0.7772	0.9077	0.7241	0.8180
$\sigma_n = 50$															
BM3D	0.7828	0.8122	0.7936	0.7433	0.7056	0.8200	0.7994	0.7722	0.7053	0.7809	0.7946	0.7068	0.8612	0.6747	0.7680
TWSC	0.7822	0.8212	0.8001	0.7570	0.7024	0.8331	0.8019	0.7837	0.7013	0.7819	0.8091	0.7029	0.8745	0.6671	0.7727
ARLLR	0.7848	0.8231	0.8008	0.7596	0.7091	0.8350	0.8059	0.7850	0.7083	0.7847	0.8199	0.7135	0.8742	0.6764	0.7772
AWGF-GPS	0.7889	0.8262	0.8058	0.7622	0.7120	0.8383	0.8073	0.7888	0.7088	0.7865	0.8160	0.7132	0.8745	0.6769	0.7790
FFDNET	0.8066	0.8244	0.8164	0.7772	0.7248	0.8509	0.8192	0.8042	0.7257	0.8031	0.7906	0.7402	0.8881	0.6910	0.7902
$\sigma_n = 70$															
BM3D	0.7427	0.7747	0.7477	0.6804	0.6548	0.7674	0.7603	0.7252	0.6526	0.7410	0.7261	0.6406	0.8120	0.6226	0.7177
TWSC	0.7421	0.7873	0.7559	0.6872	0.6498	0.7817	0.7592	0.7329	0.6420	0.7407	0.7347	0.6267	0.8337	0.6118	0.7204
ARLLR	0.7443	0.7955	0.7520	0.6943	0.6607	0.7862	0.7741	0.7399	0.6579	0.7488	0.7636	0.6524	0.8294	0.6300	0.7307
AWGF-GPS	0.7499	0.8030	0.7666	0.7001	0.6648	0.7929	0.7772	0.7470	0.6588	0.7547	0.7598	0.6527	0.8401	0.6310	0.7356
FFDNET	0.7691	0.7924	0.7693	0.7152	0.6676	0.8025	0.7835	0.7589	0.6707	0.7667	0.7211	0.6748	0.8490	0.6383	0.7414
$\sigma_n = 100$															
BM3D	0.6928	0.7203	0.6881	0.6053	0.5978	0.7021	0.7090	0.6713	0.5936	0.6898	0.6430	0.5665	0.7475	0.5650	0.6566
TWSC	0.6868	0.7280	0.6972	0.6032	0.5909	0.7151	0.7094	0.6702	0.5776	0.6904	0.6517	0.5404	0.7773	0.5490	0.6562
ARLLR	0.6966	0.7536	0.6978	0.6171	0.6048	0.7257	0.7256	0.6837	0.5981	0.7045	0.6862	0.5702	0.7767	0.5720	0.6723
AWGF-GPS	0.7038	0.7612	0.7131	0.6233	0.6110	0.7341	0.7314	0.6926	0.6011	0.7096	0.6850	0.5722	0.7840	0.5754	0.6784
FFDNET	0.4530	0.4485	0.4960	0.5046	0.4112	0.5575	0.4490	0.4642	0.4223	0.4920	0.4434	0.4264	0.4975	0.4007	0.4619

Notes: *For each test image, the corresponding denoised images with the best SSIM are marked in blue, and those with the second highest SSIM values are marked in red.

5 Conclusion

We have proposed a guided intra-patch smoothing graph filtering method for single-image denoising. Unlike the traditional AWGF, which only focuses on coefficient shrinkage, the proposed AWGF-GPS method considers more image attributes for denoising. The similarities between patches and intra-patch pixels are exploited by introducing the superpixel operation. Moreover, the guided patches from the MAP estimator provide a reliable optimal direction for the AWGF-GPS model. This provides an additional way to supervise patch restoration during graph filtering. Experiments have demonstrated that the AWGF-GPS method outperforms several state-of-the-art model-based denoising methods and is comparable with certain deep-learning methods.

Funding Statement: This work is supported by Natural Science Foundation of Jiangsu Province, China [BK20170306] and National Key R&D Program, China [2017YFC0306100]. The initials of authors who received these grants are YZ and JL, respectively. It is also supported by Fundamental Research Funds for Central Universities, China [B200202217] and Changzhou Science and Technology Program, China [CJ20200065]. The initials of author who received these grants are YT.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. Yang, Y. Xu, Y. Quan and H. Ji, "Image denoising via sequential ensemble learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 5038–5049, 2020.

[2] K. Jin and S. Wang, "Image denoising based on the asymmetric Gaussian mixture model," *Journal of Internet of Things*, vol. 2, no. 1, pp. 1–11, 2020.

[3] F. Zhang, G. Yang and J. Xue, "Hyperspectral image denoising based on low-rank coefficients and orthonormal dictionary," *Signal Processing*, vol. 177, no. 1, pp. 107738, 2020.

[4] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo *et al.,* "Deep learning on image denoising: An overview," *Neural Networks*, vol. 131, no. 11, pp. 251–275, 2020.

[5] K. Zhang, W. Zuo and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[6] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei *et al.,* "Attention-guided CNN for image denoising," *Neural Networks*, vol. 124, no. 1–2, pp. 117–129, 2020.

[7] G. Wu, S. Luo and Z. Yang, "Optimal weighted bilateral filter with dual-range kernel for Gaussian noise removal," *IET Image Processing*, vol. 14, no. 9, pp. 1840–1850, 2020.

[8] R. Verma and R. Pandey, "Adaptive selection of search region for NLM based image denoising," *Optik*, vol. 147, no. 1, pp. 151–162, 2017.

[9] X. Liu and Y. Chen, "NLTV-Gabor-based models for image decomposition and denoising," *Signal Image and Video Processing*, vol. 14, no. 2, pp. 305–313, 2020.

[10] D. Wang, Y. Xiao and Y. Gao, "Image denoising method based on NSCT bivariate model and variational bayes threshold estimation," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8927–8941, 2019.

[11] B. Khawla, R. Said and H. Abdeliah, "A wavelet denoising approach based on unsupervised learning model," *EURASIP Journal on Advances in Signal Processing*, vol. 2020, no. 36, pp. 1–26, 2020.

[12] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2090, 2007.

[13] A. A. Yahya, J. Tan, B. Su, M. Hu, Y. Wang *et al.,* "BM3D image denoising algorithm based on an adaptive filtering," *Multimedia Tools and Applications*, vol. 79, no. 4, pp. 20391–20427, 2020.

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

[15] H. G. Du, "Image denoising algorithm based on nonlocal regularization sparse representation," *IEEE Sensors Journal*, vol. 20, no. 20, pp. 11943–11950, 2020.

[16] Y. Li, G. Gui and X. Cheng, "From group sparse coding to rank minimization: A novel denoising model for low-level image restoration," *Signal Processing*, vol. 176, pp. 107655, 2020.

[17] J. Xu, L. Zhang and D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," in *Proc. European Conf. on Computer Vision*, Munich, Germany, pp. 21–38, 2018.

[18] X. Jia, X. Feng and W. Wang, "Adaptive regularizer learning for low rank approximation with application to image denoising," in *Proc. IEEE Int. Conf. on Image Processing*, Seoul, South Korea, pp. 3096–3100, 2016.

[19] Y. Chen, Y. Tang, L. Zhou, Y. Zhou, J. Zhu *et al.,* "Image denoising with adaptive weighted graph filtering," *Computers, Materials & Continua*, vol. 64, no. 2, pp. 1219–1232, 2020.

[20] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[21] Y. Fang, H. Zhang and Y. Ren, "Graph regularised sparse NMF factorisation for imagery de-noising," *IET Computer Vision*, vol. 12, no. 4, pp. 466–475, 2018.

[22] X. Zeng, W. Bian, W. Liu, J. Shen and D. Tao, "Dictionary pair learning on Grassmann manifolds for image denoising," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4556–4569, 2015.

[23] W. Waheed and D. B. H. Tay, "Graph polynomial filter for signal denoising," *IET Signal Processing*, vol. 12, no. 3, pp. 301–309, 2018.

[24] A. C. Yağan and M. T. Özgen, "Spectral graph based vertex-frequency Wiener filtering for image and graph signal denoising," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 226–240, 2020.

[25] Y. Tang, Y. Chen, N. Xu, A. Jiang and Y. Gao, "Image denoising via sparse approximation using eigenvectors of graph Laplacian," in *Proc. Visual Communications and Image Processing*, Singapore, pp. 1–4, 2015.

[26] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 755–768, 2014.

[27] F. G. Meyer and X. Shen, "Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising," *Applied and Computational Harmonic Analysis*, vol. 36, no. 2, pp. 326–334, 2014.

[28] Z. Yang, Z. Yang and D. Han, "Alternating direction method of multipliers for sparse and low-rank decomposition based on nonconvex nonsmooth weighted nuclear norm," *IEEE Access*, vol. 6, no. 1, pp. 56945–56953, 2018.

[29] Z. Yang, L. Fan, Y. Yang, Z. Yang and G. Gui, "Generalized singular value thresholding operator based nonconvex low-rank and sparse decomposition for moving object detection," *Journal of the Franklin Institute-Engineering and Applied Mathematics*, vol. 356, no. 16, pp. 10138–10154, 2019.

[30] V. Kalofolias, "How to learn a graph from smooth signals," *Artificial Intelligence and Statistics*, vol. 51, pp. 920–929, 2016.

[31] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias *et al.,* "GSPBOX: A toolbox for signal processing on graphs," *Eprint Arxiv*, vol. 61, no. 7, pp. 1644–1656, 2016.

[32] M. E. Helou and S. Süsstrunk, "Blind universal bayesian image denoising with gaussian noise level learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 4885–4897, 2020.