Multipatch Unbiased Distance Non-Local Adaptive Means With Wavelet Shrinkage

Xiaoyao Li, Yicong Zhou, *Senior Member, IEEE*, Jing Zhang, and Lianhong Wang

***Abstract*— Many existing non-local means (NLM) methods either use Euclidean distance to measure the similarity between patches, or compute weight ω_{ij} only once and keep it unchanged during the subsequent denoising iterations, or use only the structure information of the denoised image to update weight ω_{ij}. These may lead to the limited denoising performance. To address these issues, this paper proposes the non-local adaptive means (NLAM) for image denoising. NLAM treats weight ω_{ij} as an optimization variable and iteratively updates its value. We then introduce three unbiased distances, namely, pixel–pixel, patch–patch, and coupled unbiased distances. These unbiased distances are more robust to measure the image pixel/patch similarity than Euclidean distance. Using the coupled unbiased distance, we propose the unbiased distance non-local adaptive means (UD-NLAM). Because UD-NLAM uses only a single patch size to compute weight ω_{ij}, we introduce multipatch UD-NLAM (MUD-NLAM) to adapt different noise levels. To further improve denoising performance, we then propose a new denoising method called MUD-NLAM with wavelet shrinkage (MUD-NLAM-WS). Experimental results show that the proposed NLAM, UD-NLAM, and MUD-NLAM outperform existing NLM methods, and MUD-NLAM-WS achieves a better performance than the state-of-the-art denoising methods.**

***Index Terms*— image denoising, non-local means, non-local adaptive means, unbiased distance, multipatch unbiased distance non-local adaptive means, wavelet shrinkage.**

I. Introduction

WHEN we acquire or transmit an image, different types of noise are inevitably introduced. Many applications of computer vision, such as image recognition and segmentation, require image denoising as a pre-process. The goal of image denoising is to remove the noise from the original image effectively while preserving the fine details, structures and edges of the original image as much as possible. Thus, image denoising plays an important role in computer vision.

Manuscript received January 1, 2019; revised April 11, 2019 and July 1, 2019; accepted July 2, 2019. Date of publication July 19, 2019; date of current version September 12, 2019. This work was supported in part by the Science and Technology Development Fund, Macau SAR (File no. 189/2017/A3), and in part by the Research Committee at the University of Macau under Grant MYRG2016-00123-FST and Grant MYRG2018-00136-FST. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tolga Tasdizen. *(Corresponding author: Yicong Zhou.)*

X. Li is with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China, and also with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: lxyimayday@gmail.com).

Y. Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

J. Zhang and L. Wang are with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China (e-mail: zhangj@hnu.edu.cn; wanglh@hnu.edu.cn).

Digital Object Identifier 10.1109/TIP.2019.2928644

In recent years, various methods were proposed to remove Gaussian noise. Yaroslavky proposed a denoising filter [1] that considers a given neighborhood of each pixel and takes the average values of pixels which are similar in intensity. Based on the work in [1], Smith and Brandy presented the SUSAN filter [2] that computes the weights according to the spatial distance between the center pixel and the others. Using the similar idea, Sigma filter [3] and bilateral filter [4], [5] were proposed for image denoising. Dual domain image denoising (DDID) method [6], [7] alternatively uses the bilateral filter in spatial domain and wavelet shrinkage in frequency domain to obtain the optimal denoised results. Derived from DDID, progressive image denoising [8] utilizes deterministic annealing to remove the noise progressively. In addition, the Gaussian scale mixture algorithm [9] and its orientation-adapted version [10] were also proposed to remove the Gaussian noise. However, these pixel-based or local denoising methods consider only local structure information and their denoising performance is often limited.

Unlike these pixel-based or local denoising methods, the non-local means (NLM) [11], [12] introduced in 2005 takes an image patch as a processing unit, and estimates the denoised result within a non-local search area by computing the similarity between patches. Thereafter the patch-based non-local idea was widely utilized in many denoising methods. UINTA [13]–[15] is an unsupervised and adaptive filtering method that iteratively estimates the patch value by decreasing the joint entropy between the target patch and its neighbourhoods. Block-matching and 3D filtering (BM3D) [16], [17] applies collaborative filtering to the grouped similar patches to obtain an enhanced sparse representation in transform domain and to preserve the essential features of images. In 2013, Non-local Bayes (NL-Bayes) [18], [19] was proposed to calculate the denoised image patches by matrix inversion. In 2015, a novel patch-based multiscale products algorithm [20] was proposed. It also uses the patch similarity in spatial domain and multiscale products in frequency domain to obtain two initial denoised images.

Given a noisy image $\mathbf{Y} = \mathbf{X} + \mathbf{N}$, where $\mathbf{X}$ is the clean image and $\mathbf{N}$ is the Gaussian noise model, the principle of NLM is to take each image patch as a unit and calculate the weighted means of all neighbouring patches in a given search window. NLM is defined by

$$\hat{\mathbf{X}}_i = \frac{\sum_{j \in \mathbf{S}_i} \omega_{ij} \mathbf{Y}_j}{\sum_{j \in \mathbf{S}_i} \omega_{ij}}, \tag{1}$$

TABLE I
NOTATION SUMMARY

Notation	Description	Notation	Description
$\mathbf{X}$	Clean image	$\mathbf{F}_i(f), \mathbf{D}_i(f)$	Short-time Fourier transform (STFT) coefficients of i at f
$\mathbf{Y}$	Noisy image	$\mathbf{H}_i(f)$	Shrinkage factor of pixel i at f
$\mathbf{N}$	Noise model	$\mathbf{S}_i, \mathbf{S}_j$	Search windows centered at i/j
$\mathbf{G}$	Guide image	$\mathbf{S}^*$	The intersection of $\mathbf{S}_i$ and $\mathbf{S}_j$
$\hat{\mathbf{X}}$	Denoised image	s	Radius of search window
$\hat{\mathbf{Y}}, \hat{\mathbf{G}}$	Blurred layer of $\mathbf{Y}$ or $\mathbf{G}$	$\|\|\mathbf{P}\|\|$	Size of patch window $\mathbf{P}$
$\mathbf{X}(i), \mathbf{Y}(i), \mathbf{G}(i), \hat{\mathbf{X}}(i), \hat{\mathbf{Y}}(i), \hat{\mathbf{G}}(i), \hat{\mathbf{D}}(i)$	Pixel values of i in corresponding images	$p, p_1, p_2, \cdots, p_N$	Radii of patches
$\mathbf{X}_i, \mathbf{Y}_i, \mathbf{G}_i, \hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i, \hat{\mathbf{G}}_i$	Image patches centered at i of corresponding images	$\mathcal{F}_i$	Frequency window of i
$\bar{\hat{x}}_i$	The average pixel value of $\hat{\mathbf{X}}_i$	σ^2	Noise variance
$\mathbf{Y}_{i,k}, \mathbf{G}_{i,k}, \hat{\mathbf{X}}_{i,k}$	Image patches with center i and radius p_k	η_i^2	The variance of STFT coefficients
ω_{ij}	Weight between $\hat{\mathbf{X}}_i$ (or $\mathbf{Y}_i$) and $\hat{\mathbf{X}}_j$ (or $\mathbf{Y}_j$)	ε	Trade-off parameter
$\omega_{ij,k}$	Weight between $\hat{\mathbf{X}}_{i,k}$ (or $\mathbf{Y}_{i,k}$) and $\hat{\mathbf{X}}_{j,k}$ (or $\mathbf{Y}_{j,k}$)	γ	Shrinkage parameter
K_{ij}	Weight between $\mathbf{G}_i$ and $\mathbf{G}_j$	h	Trade-off parameter/Smoothing parameter
$K_{ij,k}$	Weight between $\mathbf{G}_{i,k}$ and $\mathbf{G}_{j,k}$	h_s	Spatial parameter
$\mu_{\mathbf{X}_i}, \mu_{\hat{\mathbf{X}}_i}$	The average of $\mathbf{X}_i$ or $\hat{\mathbf{X}}_i$	t	Iteration time
$\sigma^2_{\mathbf{X}_i}, \sigma^2_{\hat{\mathbf{X}}_i}$	The variance of $\mathbf{X}_i$ or $\hat{\mathbf{X}}_i$	λ_k	Weight for patch size p_k
$\sigma_{\mathbf{X}_i\hat{\mathbf{X}}_i}$	The covariance of $\mathbf{X}_i$ and $\hat{\mathbf{X}}_i$	$\mathbb{D}_{\mathrm{C}}$	Coupled unbiased distance
$\mathbb{D}_{\mathrm{U}}$	Unbiased distance	W	Normalization factor

where $\hat{\mathbf{X}}_i$ is the denoised image patch centered at pixel i, and $\mathbf{Y}_j$ is the noisy image patch centered at j. $\mathbf{S}_i$ denotes the search window with center pixel i. The weight ω_{ij} between two patches $\mathbf{Y}_i$ and $\mathbf{Y}_j$ is defined as

$$\omega_{ij} = \exp\left(-\frac{||\mathbf{Y}_i - \mathbf{Y}_j||^2}{h}\right) \tag{2}$$

where $||\cdot||$ denotes the ℓ_2 norm and h is the smoothing parameter.

Based on the framework of NLM, many improved versions were proposed successively, such as non-local Euclidean medians (NLEM) [21], improved NLEM (INLEM) [22], non-local patch regression (NLPR) [23], probabilistic NLM (PNLM) [24], NLM with local James-Stein type center pixel weights (LJS-NLM) [25] and affine NLM [26]. As seen from the optimization models of NLEM, INLEM and NLPR in Table II, they differ from weight ω_{ij} or $||\mathbf{X}_i - \mathbf{Y}_j||$. PNLM computes the probabilistic weight which reflects the distribution of the image patch distances. LJS-NLM obtains the center pixel weight by constructing the James-Stein shrinkage estimator. Affine NLM calculates the weight as the affine invariant patch similarity. These methods also aim at obtaining the average image patch in a search window, while their difference is how to calculate weight ω_{ij}. However, in order to avoid a heavy computation burden, many NLM-based methods keep weight ω_{ij} unchanged. It means that these methods compute ω_{ij} only once and keep it fixed in subsequent denoising iterations. This may lead to a bad denoising performance because the similarity between two image patches changes in each denoising iteration. Other NLM-based methods utilize only the structure information of the denoised image in each iteration. This often leads to detail loss and even over-smoothing in the denoised images.

To preserve more effective information of the original image in the iterative denoising processes, we propose four NLM-based algorithms in this paper. Our main contributions are listed below:

1. We propose the non-local adaptive means (NLAM). NLAM treats weight ω_{ij} as an optimization variable that is adaptive to the image content change, and then updates its value in each denoising iteration.
2. To better evaluate pixel/patch similarity, we introduce three unbiased distances, namely the pixel-pixel unbiased distance, patch-patch unbiased distance and coupled unbiased distance. These unbiased distances take advantage of the structure information of both the noisy image and the denoised image in each iteration. Compared with Euclidean distance, they are more robust to measure the similarity between two image pixels/patches.
3. Using the coupled unbiased distance, we propose the unbiased distance non-local adaptive means (UD-NLAM). It takes advantages of NLAM and robustness of the coupled unbiased distance.
4. Extending UD-NLAM from single patch to multipatch, we further introduce multipatch UD-NLAM (MUD-NLAM) that uses multipatch coupled unbiased distance to measure the similarity between two image patches. MUD-NLAM is able to automatically adjust the importance of denoising effect of each patch size according to different noise levels.
5. Including wavelet shrinkage, we propose the MUD-NLAM with wavelet shrinkage (MUD-NLAM-WS) that not only takes advantages of MUD-NLAM but also applies the wavelet shrinkage for preserving image details while minimizing artifacts near the edges.
6. We also provide extensive experiments to evaluate our proposed methods. The experiment results demonstrate that the proposed NLAM, UD-NLAM and MUD-NLAM show superior denoising performance over existing NLM-based

TABLE II
OPTIMIZATION MODELS

Method	Optimization Model
Mean Filter	$\hat{\mathbf{X}}(i) = \arg\min_{\mathbf{X}(i)} \sum_{j\in\mathbf{S}_i} (\mathbf{X}(i) - \mathbf{Y}(j))^2$
Gaussian Filter	$\hat{\mathbf{X}}(i) = \arg\min_{\mathbf{X}(i)} \sum_{j\in\mathbf{S}_i} \omega_{ij} (\mathbf{X}(i) - \mathbf{Y}(j))^2$
NLM [22]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} \omega_{ij} \|\mathbf{X}_i - \mathbf{Y}_j\|^2$
NLEM [21]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} \omega_{ij} \|\mathbf{X}_i - \mathbf{Y}_j\|$
INLEM [22]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} \sqrt{\omega_{ij}} \|\mathbf{X}_i - \mathbf{Y}_j\|$
NLPR [23]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} \omega_{ij} \|\mathbf{X}_i - \mathbf{Y}_j\|^P$
PNLM [24]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} f_{ij} \|\mathbf{X}_i - \mathbf{Y}_j\|^2$

methods, and MUD-NLAM-WS shows the competitive denoising performance with BM3D and outperforms other algorithms in terms of quantitative evaluations and visual comparisons.

Our initial works were published in conference papers [27] and [28]. Compared with the conference papers, we propose two new denoising methods, i.e. MUD-NLAM and MUD-NLAM-WS, and perform comprehensive experiments for algorithm validation.

This paper is organized as follows. Section II will briefly review the optimization models of several filters and NLM-based methods. Section III will introduce NLAM. Section IV will define three unbiased distances and then present the proposed UD-NLAM. Section V will propose MUD-NLAM. Section VI will describe the new denoising method MUD-NLAM-WS in detail. Section VII will show the experiment results in both quantitative evaluations and visual effects. Finally, Section VIII will summarize the paper.

For easy reading, Table I summarizes the main notations used in this paper.

II. OPTIMIZATION MODELS

Due to the fact that NLM was proposed to find the optimal center patch in a search window, we can convert it into an optimization problem. The optimization models of five NLM-based methods are listed in Table II. They are traditional NLM, NLEM, INLEM, NLPR and PNLM. In addition, we also include two filters in Table II. They are the mean filter and Gaussian filter. In Table II, $\hat{\mathbf{X}}(i)$, $\mathbf{X}(i)$, $\mathbf{Y}(j)$ denote the values of pixel i/j in the denoised image $\hat{\mathbf{X}}$, clean image $\mathbf{X}$ and noisy image $\mathbf{Y}$, respectively. $\hat{\mathbf{X}}_i$, $\mathbf{X}_i$, $\mathbf{Y}_j$ are patches centered at the pixel i/j in $\hat{\mathbf{X}}$, $\mathbf{X}$ and $\mathbf{Y}$, respectively.

Mean filter aims at calculating the average value of all pixels within a given window and set it to the target pixel value. Gaussian filter can be considered as a weighted mean filter and it calculates the average in a Gaussian weighted window. NLM [11] uses an image patch as a processing unit and calculates the weighted patch in a search window. By changing the distance term $\|\mathbf{X}_i - \mathbf{Y}_j\|^2$ into $\|\mathbf{X}_i - \mathbf{Y}_j\|$, NLEM [21] obtains better denoising effect than traditional NLM. Based on NLEM, INLEM [22] further modifies weight ω_{ij} into $\sqrt{\omega_{ij}}$. NLPR [23] keeps the same weight style as NLM while replacing the ℓ^2 norm in NLM with ℓ^P norm. PNLM [24] uses the distribution of patch differences to calculate the probabilistic weight f_{ij}.

However, these denoising methods calculate weight ω_{ij} only once and keep it unchanged in the subsequent denoising iterations. This results in less computation cost but worse denoising performance because the similarity between two patches changes during iterative denoising processes.

III. NLAM

In Table II, the denoising methods update only $\hat{\mathbf{X}}(i)$ or $\hat{\mathbf{X}}_i$ and keep weight ω_{ij} fixed in each denoising iteration. This is improper because the similarity between two image patches will change in each denoising result. To solve this problem, we propose the non-local adaptive means (NLAM), which makes weight ω_{ij} to be iteratively adaptive to image content change. NLAM treats weight ω_{ij} as an optimization variable and calculates its value in each denoising iteration. Thus, in terms of the non-local functional in [29], [30], the optimization model of NLAM can be defined as

$$\begin{aligned}\{\hat{\mathbf{X}}_i, \omega_{ij}\} = \arg\min_{\mathbf{X}_i,\omega_{ij}} & \sum_{j\in\mathbf{S}_i} \omega_{ij} \|\mathbf{X}_i - \mathbf{Y}_j\|^2 \\ & + h \sum_{j\in\mathbf{S}_i} \omega_{ij} \log \omega_{ij} \\ s.t.\ \omega_{ij} \geq 0, & \sum_{j\in\mathbf{S}_i} \omega_{ij} = 1. \end{aligned} \tag{3}$$

where $\sum_{j\in\mathbf{S}_i} \omega_{ij} \|\mathbf{X}_i - \mathbf{Y}_j\|^2$ and $\sum_{j\in\mathbf{S}_i} \omega_{ij} \log \omega_{ij}$ are data and regularization terms, respectively. $h \geq 0$ is a trade-off parameter.

To solve the optimization model in Eq. (3), we first fix $\hat{\mathbf{X}}_i$ and solve ω_{ij} by setting the derivative with respect to ω_{ij} to zero, and then fix ω_{ij} to calculate $\hat{\mathbf{X}}_i$. Given the initial settings of $\hat{\mathbf{X}}^{(0)} = \mathbf{Y}$ and $t = 0$, after $t+1$ denoising iterations, we alternatively calculate ω_{ij} and $\hat{\mathbf{X}}_i$ as

$$W_i^{(t)} = \sum_{j\in\mathbf{S}_i} \exp\left(-\frac{\|\hat{\mathbf{X}}_i^{(t)} - \mathbf{Y}_j\|^2}{h}\right). \tag{4}$$

$$\omega_{ij}^{(t+1)} = \frac{1}{W_i^{(t)}} \cdot \exp\left(-\frac{\|\hat{\mathbf{X}}_i^{(t)} - \mathbf{Y}_j\|^2}{h}\right), \tag{5}$$

$$\hat{\mathbf{X}}_i^{(t+1)} = \frac{\sum_{j\in\mathbf{S}_i} \omega_{ij}^{(t+1)} \mathbf{Y}_j}{\sum_{j\in\mathbf{S}_i} \omega_{ij}^{(t+1)}}, \tag{6}$$

where $W_i^{(t)}$ denotes the normalization factor. The iterative denoising processes of NLAM are shown in Algorithm 1.

IV. UD-NLAM

This section introduces three distances to measure the similarity between pixels or patches, namely pixel-pixel unbiased distance, patch-patch unbiased distance and coupled unbiased distance. Using the coupled unbiased distance, the unbiased distance non-local adaptive means (UD-NLAM) is then proposed.

Algorithm 1 NLAM

Input: Noisy image $\mathbf{Y}$; Radius of patch p, radius of search window s and smoothing parameter h;
Output: Denoised image $\hat{\mathbf{X}}$;
1: Initialize $\hat{\mathbf{X}}^{(0)} = \mathbf{Y}$ and $t = 0$;
2: **while** $||\hat{\mathbf{X}}^{(t+1)} - \hat{\mathbf{X}}^{(t)}||_2 > \tau$ **do**
3: **for** each pixel located at i **do**
4: Extract a patch $\hat{\mathbf{X}}_i^{(t)}$ with center i;
5: Calculate $\omega_{ij}^{(t+1)}$ ($j \in \mathbf{S}_i$) by Eq. (5);
6: Calculate $\hat{\mathbf{X}}_i^{(t+1)}$ by Eq. (6);
7: **end for**
8: $t = t + 1$;
9: **end while**
10: **return** $\hat{\mathbf{X}}$

A. Unbiased Distances

1) Pixel-Pixel Unbiased Distance: Given two noisy pixels $\mathbf{Y}(i)$ and $\mathbf{Y}(j)$ centered at i and j of noisy image $\mathbf{Y}$, and their corresponding clean pixels $\mathbf{X}(i)$ and $\mathbf{X}(j)$ from clean image $\mathbf{X}$, the relationship between $\big(\mathbf{Y}(i)-\mathbf{Y}(j)\big)^2$ and $\big(\mathbf{X}(i)-\mathbf{X}(j)\big)^2$ can be described as the following equation [11], [12], [31],

$$\mathbf{E}\Big[\big(\mathbf{Y}(i)-\mathbf{Y}(j)\big)^2\Big] = \big(\mathbf{X}(i)-\mathbf{X}(j)\big)^2 + 2\sigma^2. \tag{7}$$

where $\mathbf{E}[\cdot]$ is expectation calculator and σ^2 is the noise variance.

Inspired by Eq. (7), the squared pixel-pixel unbiased distance between $\mathbf{Y}(i)$ and $\mathbf{Y}(j)$ is defined as

$$\mathbb{D}_{\mathrm{U}}^2\big(\mathbf{Y}(i),\mathbf{Y}(j)\big) = \big(\mathbf{Y}(i)-\mathbf{Y}(j)\big)^2 - 2\sigma^2. \tag{8}$$

This equation shows that the unbiased distance conveys more similarity information between two pixels than Euclidean distance by removing the bias caused by noise.

Similarly, if we have $\hat{\mathbf{X}}(i) = \sum_{l\in\mathbf{S}_i} \omega_{il}\mathbf{Y}(l)$, $\sum_{l\in\mathbf{S}_i} \omega_{il} = 1$ and $j \in \mathbf{S}_i$, we can define the squared unbiased distance between the denoised image pixel $\hat{\mathbf{X}}(i)$ and the noisy image pixel $\mathbf{Y}(j)$ as

$$\mathbb{D}_{\mathrm{U}}^2\big(\hat{\mathbf{X}}(i),\mathbf{Y}(j)\big) = \big(\hat{\mathbf{X}}(i)-\mathbf{Y}(j)\big)^2 - \Big(\sum_{l\in\mathbf{S}_i}\omega_{il}^2 - 2\omega_{ij} + 1\Big)\sigma^2. \tag{9}$$

More generally, the squared unbiased distance between $\hat{\mathbf{X}}(i)$ and $\hat{\mathbf{X}}(j)$ is

$$\begin{aligned}\mathbb{D}_{\mathrm{U}}^2\big(\hat{\mathbf{X}}(i),\hat{\mathbf{X}}(j)\big) &= \big(\hat{\mathbf{X}}(i)-\hat{\mathbf{X}}(j)\big)^2 \\ &\quad - \Big(\sum_{l\in\mathbf{S}_i}\omega_{il}^2 + \sum_{q\in\mathbf{S}_j}\omega_{jq}^2 - 2\sum_{n\in\mathbf{S}^*}\omega_{in}\omega_{jn}\Big)\sigma^2,\end{aligned} \tag{10}$$

where $\hat{\mathbf{X}}(i) = \sum_{l\in\mathbf{S}_i}\omega_{il}\mathbf{Y}(l)$, $\hat{\mathbf{X}}(j) = \sum_{q\in\mathbf{S}_j}\omega_{iq}\mathbf{Y}(q)$ and $\mathbf{S}^* = \mathbf{S}_i \bigcap \mathbf{S}_j$. $\mathbf{S}_i$ and $\mathbf{S}_j$ are two search windows centered at pixel i and j, respectively.

2) Patch-Patch Unbiased Distance: Given two noisy image patches $\mathbf{Y}_i$ and $\mathbf{Y}_j$, we add up all the squared pixel-pixel unbiased distances within the patch window $\mathbf{P}$ to calculate the squared patch-patch unbiased distance as

$$\begin{aligned}\mathbb{D}_{\mathrm{U}}^2\big(\mathbf{Y}_i,\mathbf{Y}_j\big) &= \sum_{p\in\mathbf{P}}\mathbb{D}_{\mathrm{U}}^2\big(\mathbf{Y}_i(p),\mathbf{Y}_j(p)\big) \\ &= ||\mathbf{Y}_i - \mathbf{Y}_j||^2 - 2||\mathbf{P}||\sigma^2,\end{aligned} \tag{11}$$

where $||\mathbf{P}||$ denotes the number of pixels within the patch window $\mathbf{P}$. $\mathbf{Y}_i(p)$ and $\mathbf{Y}_j(p)$ are the values of the pth pixels in $\mathbf{Y}_i$ and $\mathbf{Y}_j$, respectively.

Similarly, we can calculate the squared unbiased distance between the denoised image patch $\hat{\mathbf{X}}_i$ and the noisy patch $\mathbf{Y}_j$ as

$$\begin{aligned}\mathbb{D}_{\mathrm{U}}^2\big(\hat{\mathbf{X}}_i,\mathbf{Y}_j\big) &= \sum_{p\in\mathbf{P}}\mathbb{D}_{\mathrm{U}}^2\big(\hat{\mathbf{X}}_i(p),\mathbf{Y}_j(p)\big) \\ &= ||\hat{\mathbf{X}}_i - \mathbf{Y}_j||^2 - ||\mathbf{P}||\Big(\sum_{l\in\mathbf{S}_i}\omega_{il}^2 - 2\omega_{ij} + 1\Big)\sigma^2,\end{aligned} \tag{12}$$

and the squared unbiased distance between two denoised image patches $\hat{\mathbf{X}}_i$ and $\hat{\mathbf{X}}_j$ as

$$\begin{aligned}\mathbb{D}_{\mathrm{U}}^2\big(\hat{\mathbf{X}}_i,\hat{\mathbf{X}}_j\big) &= \sum_{p\in\mathbf{P}}\mathbb{D}_{\mathrm{U}}^2\big(\hat{\mathbf{X}}_i(p),\hat{\mathbf{X}}_j(p)\big) \\ &= ||\hat{\mathbf{X}}_i - \hat{\mathbf{X}}_j||^2 - ||\mathbf{P}||\Big(\sum_{l\in\mathbf{S}_i}\omega_{il}^2 + \sum_{q\in\mathbf{S}_j}\omega_{jq}^2 \\ &\quad - 2\sum_{n\in\mathbf{S}^*}\omega_{in}\omega_{jn}\Big)\sigma^2,\end{aligned} \tag{13}$$

where $\hat{\mathbf{X}}_i = \sum_{l\in\mathbf{S}_i}\omega_{il}\mathbf{Y}_l$, $\hat{\mathbf{X}}_j = \sum_{q\in\mathbf{S}_j}\omega_{iq}\mathbf{Y}_q$ and $\mathbf{S}^* = \mathbf{S}_i \bigcap \mathbf{S}_j$.

3) Coupled Unbiased Distance: Using the patch-patch unbiased distance, we can define the squared coupled unbiased distance between $\hat{\mathbf{X}}_i$ (or $\mathbf{Y}_i$) and $\hat{\mathbf{X}}_j$ (or $\mathbf{Y}_j$) as

$$\mathbb{D}_{\mathrm{C}}^2\big(\hat{\mathbf{X}}_i,\hat{\mathbf{X}}_j\big) = \varepsilon\cdot\max\big[0,\mathbb{D}_{\mathrm{U}}^2(\hat{\mathbf{X}}_i,\hat{\mathbf{X}}_j)\big] + (1-\varepsilon)\cdot\big(\bar{\hat{x}}_i - \bar{\hat{x}}_j\big)^2. \tag{14}$$

where ε is the trade-off parameter and $0 \le \varepsilon \le 1$. $\bar{\hat{x}}_i$ and $\bar{\hat{x}}_j$ are average pixel values of $\hat{\mathbf{X}}_i$ and $\hat{\mathbf{X}}_j$.

Since $\mathbb{D}_{\mathrm{U}}^2(\hat{\mathbf{X}}_i,\hat{\mathbf{X}}_j)$ may be negative, we apply $\max\big[0,\mathbb{D}_{\mathrm{U}}^2(\hat{\mathbf{X}}_i,\hat{\mathbf{X}}_j)\big]$ in Eq. (14) to guarantee that the patch-patch unbiased distance is not less than zero. Compared with Euclidean distance, the coupled unbiased distance contains more structure information because it calculates not only patch-patch distance $\big(\text{i.e. } \max\big[0,\mathbb{D}_{\mathrm{U}}^2(\hat{\mathbf{X}}_i,\hat{\mathbf{X}}_j)\big]\big)$ but also pixel-pixel distance $\big(\text{i.e. } \big(\bar{\hat{x}}_i - \bar{\hat{x}}_j\big)^2\big)$. In addition, the coupled unbiased distance releases the similarity requirement such that we can find more similar neighboring patches $\hat{\mathbf{X}}_j$ satisfying $\mathbb{D}_{\mathrm{C}}^2\big(\hat{\mathbf{X}}_i,\hat{\mathbf{X}}_j\big) = 0$ than that satisfying $||\hat{\mathbf{X}}_i - \hat{\mathbf{X}}_j||^2 = 0$.

B. UD-NLAM

Using the coupled unbiased distances, this subsection introduces the unbiased distance non-local adaptive means

Algorithm 2 UD-NLAM

Input: Noisy image $\mathbf{Y}$; Radius of patch p, radius of search window s and parameters σ^2, h, h_s, ε;
Output: Denoised image $\hat{\mathbf{X}}$;
1: Initialize $\hat{\mathbf{X}}^{(0)} = \mathbf{Y}$ and $t = 0$;
2: **while** $||\hat{\mathbf{X}}^{(t+1)} - \hat{\mathbf{X}}^{(t)}||_2 > \tau$ **do**
3: **for** each pixel located at i **do**
4: Extract a patch $\hat{\mathbf{X}}_i^{(t)}$ with center i;
5: Calculate $\omega_{ij}^{(t+1)}$ ($j \in \mathbf{S}_i$) by Eq. (17);
6: Calculate $\hat{\mathbf{X}}_i^{(t+1)}$ by Eq. (18);
7: **end for**
8: $t = t + 1$;
9: **end while**
10: **return** $\hat{\mathbf{X}}$

(UD-NLAM). Its optimization model is given as

$$\{\hat{\mathbf{X}}_i, \omega_{ij}\} = \arg\min_{\mathbf{X}_i, \omega_{ij}} \sum_{j \in \mathbf{S}_i} \omega_{ij} \mathbb{D}_{\mathrm{C}}^2(\mathbf{X}_i, \mathbf{Y}_j) + h \sum_{j \in \mathbf{S}_i} \omega_{ij} \log \omega_{ij}$$
$$s.t.\ \omega_{ij} \geq 0, \quad \sum_{j \in \mathbf{S}_i} \omega_{ij} = 1. \tag{15}$$

Firstly, we replace the Euclidean distance $||\mathbf{X}_i - \mathbf{Y}_j||^2$ of NLAM in Eq. (3) with the coupled unbiased distance $\mathbb{D}_{\mathrm{C}}^2(\mathbf{X}_i, \mathbf{Y}_j)$ that is robust to measure the similarity between two image patches. Secondly, keeping the content-adaptive concept of NLAM, UD-NLAM also considers weight ω_{ij} as an optimization variable and updates its value in each denoising iteration. Similar to NLAM, given the initial state of $\hat{\mathbf{X}}_i^{(0)} = \mathbf{Y}_i$, we use the following two equations to update ω_{ij} and $\hat{\mathbf{X}}_i$ alternatively:

$$W_i^{(t)} = \sum_{j \in \mathbf{S}_i} \exp\left(-\frac{\mathbb{D}_{\mathrm{C}}^2(\hat{\mathbf{X}}_i^{(t)}, \mathbf{Y}_j)}{h}\right) \cdot H_{ij} \tag{16}$$

$$\omega_{ij}^{(t+1)} = \frac{1}{W_i^{(t)}} \cdot \exp\left(-\frac{\mathbb{D}_{\mathrm{C}}^2(\hat{\mathbf{X}}_i^{(t)}, \mathbf{Y}_j)}{h}\right) \cdot H_{ij} \tag{17}$$

$$\hat{\mathbf{X}}_i^{(t+1)} = \frac{\sum_{j \in \mathbf{S}_i} \omega_{ij}^{(t+1)} \mathbf{Y}_j}{\sum_{j \in \mathbf{S}_i} \omega_{ij}^{(t+1)}}, \tag{18}$$

where the spatial kernel $H_{ij} = \exp\left(-\frac{(i-j)^2}{h_s}\right)$ where h_s is the spatial parameter. $W_i^{(t)}$ is the normalization factor. $\hat{\mathbf{X}}_i^{(t+1)}$ and $\omega_{ij}^{(t+1)}$ are the denoised image patch $\hat{\mathbf{X}}_i$ and weight ω_{ij} after $t+1$ iterations. The procedures of UD-NLAM are shown in Algorithm 2.

C. Comparison of UD-NLAM, NLAM and NLM

First, in terms of the optimization models, NLM takes only $\mathbf{X}_i$ as an optimization variable and calculates weight ω_{ij} once and keeps it fixed in the subsequent denoising iterations. NLAM treats both $\mathbf{X}_i$ and ω_{ij} as optimization variables and updates their values iteratively. UD-NLAM inherits the adaptive concept of NLAM, and introduces the coupled unbiased distance instead of Euclidean distance to measure the image patch similarity.

Second, according to Eqs. (8)-(14), the unbiased distances are based on the adaptive ω_{ij}. If ω_{ij} is fixed, we will calculate only the distance between two noisy patches by Eqs. (8) and (11). UD-NLAM will revert back to NLM. If we use the adaptive ω_{ij} and Euclidean distance, instead of unbiased distances, UD-NLAM will revert back to NLAM.

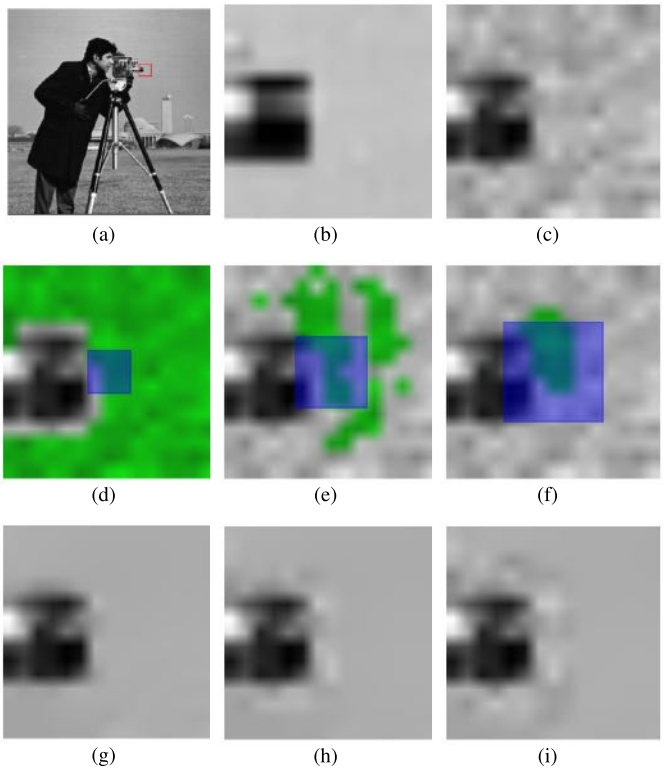

Fig. 1. Denoised results of the *cameraman* image ($\sigma = 20$) using UD-NLAM with different patch sizes: (a) clean image, and zoom-in region of (b) clean image, and (c) noisy image; similar neighbouring patches (green region) in the search window using (d) 3×3, (e) 5×5, and (f) 7×7 patches (blue square), individually; and zoom-in denoised results with (g) 3×3, (h) 5×5, and (i) 7×7 patches, respectively.

V. MUD-NLAM

UD-NLAM uses a single patch size to calculate ω_{ij} on each noise level. Its denoising performance changes dramatically for different noise levels. Here, we provide two examples of UD-NLAM. Their denoised results are obtained by using patch sizes 3×3, 5×5 and 7×7, respectively. Fig. 1 shows the zoom-in version of the denoised results of the *cameraman* image using UD-NLAM when noise level $\sigma = 20$. The second row of Fig. 1 shows the center patches (blue squares) in the search window and their similar neighbouring patches (green regions). As can be seen, there are less similar neighbouring patches when the patch size increases. Their denoised results are shown in the third row of Fig. 1. When the patch size is 5×5 (Fig. 1(h)) or 7×7 (Fig. 1(i)), there remains noise around edges of the object. The results indicate that a smaller patch

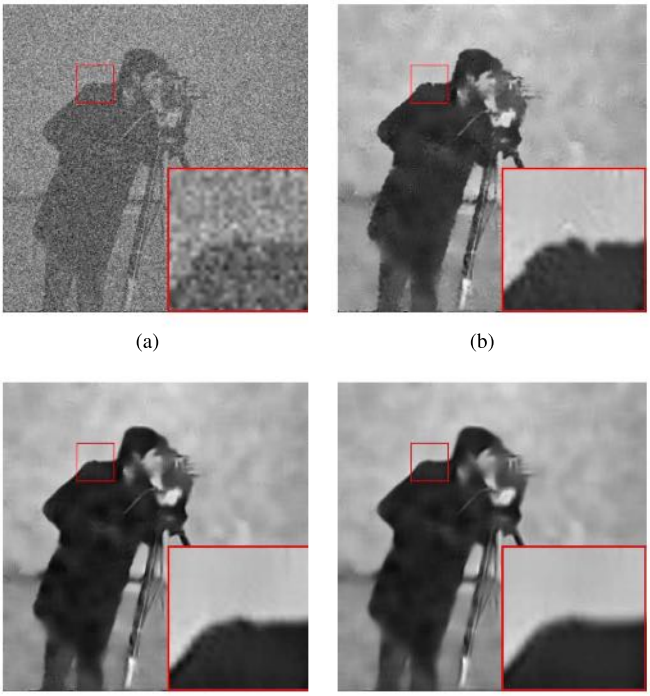

Fig. 2. Denoised results of the *cameraman* image ($\sigma = 80$) using UD-NLAM with different patch sizes: (a) noisy image; and denoised results with (b) 3×3, (c) 5×5, and (d) 7×7 patches, respectively.

is more appropriate in removing low-level noise than a larger patch. Fig. 2 shows the denoised results of the *cameraman* image using UD-NLAM when $\sigma = 80$. Observing from the denoised result with 3×3 patch in Fig. 2(b), the hollow formed by noise is wrongly preserved as a signal. Many obvious ringing artifacts are introduced on the background. The denoised results obtained by using 5×5 (Fig. 2(c)) and 7×7 (Fig. 2(d)) patches have better visual effects because a larger patch keeps more structure information and a smaller patch is more sensitive to the noise pattern. Therefore, the large patch achieves better denoising effect than the small patch on high noise levels.

Motivated by these results, we further propose multipatch UD-NLAM (MUD-NLAM) to combine the advantages of different patch sizes with UD-NLAM. Given a set of patch radii $\mathbf{K} = \{p_1, p_2, \cdots, p_N\}$, we firstly calculate $\omega_{ij,k}$ for each $p_k \in \mathbf{K}$ as

$$\omega_{ij,k}^{(t+1)} = \frac{1}{W_{i,k}^{(t)}} \cdot \exp\left(-\frac{\mathbb{D}_{\mathrm{C}}^2\left(\hat{\mathbf{X}}_{i,k}^{(t)}, \mathbf{Y}_{j,k}\right)}{h}\right) \cdot H_{ij} \tag{19}$$

where $\hat{\mathbf{X}}_{i,k}^{(t)}$ and $\mathbf{Y}_{j,k}$ are patches with patch radius p_k and center pixel i. Then Eq. (17) can be modified as

$$\omega_{ij}^{(t+1)} = \sum_{k=1}^{N} \lambda_k \omega_{ij,k}^{(t+1)}, \tag{20}$$

where $\sum_{k=1}^{N} \lambda_k = 1$. In our experiments, $\lambda_k = c\sigma^a$, where σ denotes the noise level and c is a positive constant. For a small patch size, we set $a < 0$ such that λ_k decreases with the increase of noise level. For a large patch size, the change of λ_k is opposite to that for the small patch when setting $a > 0$. This can also be referenced in real applications.

Algorithm 3 MUD-NLAM

Input: Noisy image $\mathbf{Y}$; Parameters $\mathbf{K} = \{p_1, p_2, \cdots, p_N\}$, s, σ^2, h, h_s, ε;
Output: Denoised image $\hat{\mathbf{X}}$; Weight ω_{ij};

```
1: Initialize X̂^(0) = Y and t = 0;
2: while ||X̂^(t+1) − X̂^(t)||_2 > τ do
3:    for each pixel located at i do
4:       for each patch radius p_k ∈ K do
5:          Extract a patch X̂_{i,k}^(t) with radius p_k and center i;
6:          Calculate ω_{ij,k}^(t+1) (j ∈ S_i) by Eq. (19);
7:       end for
8:       Calculate ω_ij^(t+1) (j ∈ S_i) by Eq. (20);
9:       Calculate X̂_i^(t+1) by Eq. (18);
10:   end for
11:   t = t + 1;
12: end while
13: return X̂, ω_ij
```

The detail processes of MUD-NLAM are presented in Algorithm 3. Compared with UD-NLAM in Algorithm 2, we add steps 4-7 to calculate $\omega_{ij,k}$ for multipatch denoising.

VI. MUD-NLAM-WS

Based on MUD-NLAM, this section proposes a new denoising method, named MUD-NLAM with wavelet shrinkage (MUD-NLAM-WS). The denoising processes of MUD-NLAM-WS are illustrated in Fig. 3. In the proposed MUD-NLAM-WS, we use MUD-NLAM to generate a guide image, and then alternatively use MUD-NLAM and wavelet shrinkage [6], [32] to obtain the final denoised image.

The detail procedures of MUD-NLAM-WS are shown in Algorithm 4. In the stage of guide image generation, we apply MUD-NLAM to preprocess the noisy image $\mathbf{Y}$ and obtain the denoised image $\hat{\mathbf{X}}$ and weight ω_{ij} by Eqs. (19) and (20). Then, we set the denoised image $\hat{\mathbf{X}}$ as the initial guide image $\mathbf{G}^{(0)}$, and use the following equations to update $\mathbf{G}$ and $\hat{\mathbf{X}}$. In the $(t+1)$th iteration, we first calculate weight $K_{ij}^{(t+1)}$ as

$$K_{ij,k}^{(t+1)} = \frac{1}{W_{i,k}^{(t)}} \cdot \exp\left(-\frac{\mathbb{D}_{\mathrm{C}}^2\left(\mathbf{G}_{i,k}^{(t)}, \mathbf{G}_{j,k}^{(t)}\right)}{h}\right) \cdot H_{ij}, \tag{21}$$

$$K_{ij}^{(t+1)} = \sum_{k=1}^{N} \lambda_k K_{ij,k}^{(t+1)}, \tag{22}$$

where N denotes the size of the set of patch radii $\mathbf{K}$. Then we obtain $\hat{\mathbf{G}}_i^{(t+1)}$ and $\hat{\mathbf{Y}}_i^{(t+1)}$ as

$$\hat{\mathbf{G}}_i^{(t+1)} = \frac{\sum\limits_{j \in \mathbf{S}_i} K_{ij}^{(t+1)} \mathbf{G}_j^{(t)}}{\sum\limits_{j \in \mathbf{S}_i} K_{ij}^{(t+1)}}, \tag{23}$$

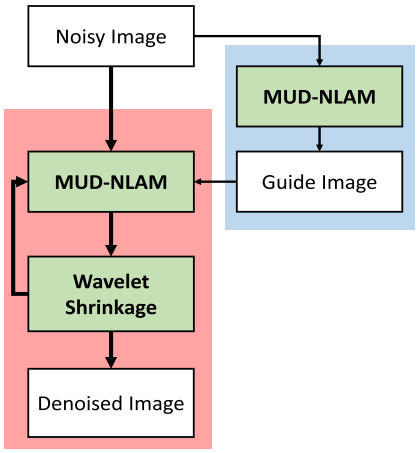

Fig. 3. The flow chart of MUD-NLAM-WS. The blue block denotes the stage of guide image generation and the red block denotes the stage of image denoising.

$$\hat{\mathbf{Y}}_i^{(t+1)} = \frac{\sum\limits_{j\in\mathbf{S}_i} K_{ij}^{(t+1)}\mathbf{Y}_j}{\sum\limits_{j\in\mathbf{S}_i} K_{ij}^{(t+1)}}, \tag{24}$$

where $\hat{\mathbf{G}}$ and $\hat{\mathbf{Y}}$ are the blurred layers of $\mathbf{G}$ and $\mathbf{Y}$, respectively.

Next, we obtain the short-time Fourier transform (STFT) [33] coefficients $\mathbf{F}_i^{(t+1)}(f)$ and $\mathbf{D}_i^{(t+1)}(f)$ as

$$K_{ij}^{(t+1)} \cdot \left(\mathbf{G}^{(t)}(j) - \hat{\mathbf{G}}^{(t+1)}(j)\right) \xrightarrow[j\in\mathbf{S}_i]{STFT} \mathbf{F}_i^{(t+1)}(f), \tag{25}$$

$$K_{ij}^{(t+1)} \cdot \left(\mathbf{Y}(j) - \hat{\mathbf{Y}}^{(t+1)}(j)\right) \xrightarrow[j\in\mathbf{S}_i]{STFT} \mathbf{D}_i^{(t+1)}(f), \tag{26}$$

where $\mathbf{G}^{(t)}(j), \hat{\mathbf{G}}^{(t+1)}(j), \mathbf{Y}(j), \hat{\mathbf{Y}}^{(t+1)}(j)$ are image pixels centered at j.

The frequency window $\mathcal{F}_i$ is of the same size as the search window $\mathbf{S}_i$ and $f \in \mathcal{F}_i$. Using $\mathbf{F}_i^{(t+1)}(f)$, we calculate the shrinkage factor $\mathbf{H}_i^{(t+1)}(f)$ as

$$\mathbf{H}_i^{(t+1)}(f) = \exp\left(-\frac{\gamma\,\eta_i^2}{|\mathbf{F}_i^{(t+1)}(f)|^2}\right), \tag{27}$$

where the variance $\eta_i^2 = \sigma^2 \sum\limits_{j\in\mathbf{S}_i}\left(K_{ij}^{(t+1)}\right)^2$ and γ denotes the shrinkage parameter.

Then we utilize the inverse short-time Fourier transform (ISTFT) and obtain the detail layer $\hat{\mathbf{D}}$ as

$$\mathbf{H}_i^{(t+1)}(f)\mathbf{D}_i^{(t+1)}(f) \xrightarrow[f\in\mathcal{F}_i]{ISTFT} \hat{\mathbf{D}}^{(t+1)}(i), \tag{28}$$

where $\hat{\mathbf{D}}^{(t+1)}(i)$ is the pixel value of i in $\hat{\mathbf{D}}^{(t+1)}$.

Finally, we update the denoised image as $\hat{\mathbf{X}}^{(t+1)} = \hat{\mathbf{Y}}^{(t+1)} + \hat{\mathbf{D}}^{(t+1)}$ and then the guide image as $\mathbf{G}^{(t+1)} = \hat{\mathbf{X}}^{(t+1)}$.

As shown in Fig. 3, we call the first stage of MUD-NLAM-WS as guide image generation (i.e. the blue block) and the second stage as image denoising (i.e. the red block). The detail steps of MUD-NLAM-WS are shown in Algorithm 4.

Algorithm 4 MUD-NLAM-WS

Input: Noisy image $\mathbf{Y}$; Parameters $\mathbf{K} = \{p_1, p_2, \cdots, p_N\}$, s, σ^2, h, h_s, ε, γ;
Output: Denoised image $\hat{\mathbf{X}}$;
1: Use MUD-NLAM and obtain $\hat{\mathbf{X}}$ and ω_{ij};
2: Initialize $\mathbf{G}^{(0)} = \hat{\mathbf{X}}$ and $t = 0$;
3: **while** $||\hat{\mathbf{X}}^{(t+1)} - \hat{\mathbf{X}}^{(t)}||_2 > \tau$ **do**
4: **for** each pixel located at i **do**
5: **for** each patch radius $p_k \in \mathbf{K}$ **do**
6: Extract a patch $\mathbf{G}_{i,k}^{(t)}$ with radius p_k and center i;
7: Calculate $K_{ij,k}^{(t+1)}$ $(j \in \mathbf{S}_i)$ by Eq. (21);
8: **end for**
9: Calculate $K_{ij}^{(t+1)}$ $(j \in \mathbf{S}_i)$ by Eq. (22);
10: Calculate $\hat{\mathbf{G}}_i^{(t+1)}$ and $\hat{\mathbf{Y}}_i^{(t+1)}$ by Eqs. (23)-(24);
11: Use STFT and obtain coefficients by Eqs. (25)-(26);
12: Calculate shrinkage factors $\mathbf{H}_i^{(t+1)}(f)$ by Eq. (27);
13: Obtain ISTFT result $\hat{\mathbf{D}}^{(t+1)}(i)$ by Eq. (28);
14: **end for**
15: $\hat{\mathbf{X}}^{(t+1)} = \hat{\mathbf{Y}}^{(t+1)} + \hat{\mathbf{D}}^{(t+1)}$;
16: $\mathbf{G}^{(t+1)} = \hat{\mathbf{X}}^{(t+1)}$;
17: $t = t + 1$;
18: **end while**
19: **return** $\hat{\mathbf{X}}$

The guide image $\mathbf{G}$ is a pre-denoised image that is used to guide the noisy image $\mathbf{Y}$ to go through the iterative denoising processes. $\hat{\mathbf{X}}$ denotes the denoised result after each denoising iteration. They are two different images in MUD-NLAM-WS. In the stage of guide image generation in Fig. 3, the guide image $\mathbf{G}$ is initialized as the denoised result of applying MUD-NLAM once. $\mathbf{G}$ can provide more accurate structural information than the noisy image $\mathbf{Y}$ itself when computing weight $K_{ij,k}$. Besides, the parameter settings of MUD-NLAM in the stage of guide image generation are different from these in the stage of image denoising.

VII. Experiments

A. Quality Measures

To evaluate the performance of the denoising methods, we use three image quality measures including the peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [34] index and method noise.

1) PSNR: PSNR measures the intensity similarity between the clean image $\mathbf{X}$ and the denoised image $\hat{\mathbf{X}}$. The higher PSNR value indicates the more intensity similarity of $\mathbf{X}$ and $\hat{\mathbf{X}}$. Its calculation equation is

$$PSNR(\mathbf{X}, \hat{\mathbf{X}}) = -10 \cdot \log_{10}\left(\frac{\frac{1}{N_1 N_2}||\mathbf{X} - \hat{\mathbf{X}}||_2^2}{MAX_{\mathbf{X}}^2}\right), \tag{29}$$

where $\mathbf{X}$, $\hat{\mathbf{X}}$ are of size $N_1 \times N_2$ and $MAX_{\mathbf{X}}$ is the maximum pixel value of $\mathbf{X}$.

2) SSIM: SSIM [34] measures the structure between the clean image $\mathbf{X}$ and the denoised image $\hat{\mathbf{X}}$. The larger SSIM

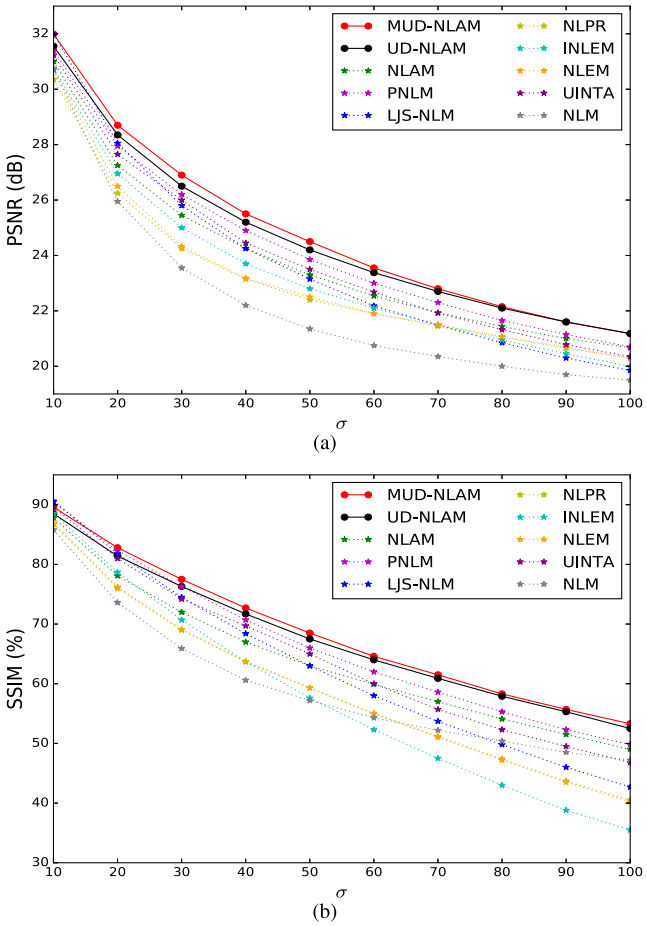

Fig. 4. Average values of PSNR and SSIM on the test images: (a) average PSNR; (b) average SSIM.

value means the more similarity between $\mathbf{X}$ and $\hat{\mathbf{X}}$, and thus the better denoising performance. Its calculation equation is

$$SSIM(\mathbf{X},\hat{\mathbf{X}}) = \frac{1}{N_1 N_2}\sum_{i=1}^{N_1 N_2} \times \frac{(2\mu_{\mathbf{X}_i}\mu_{\hat{\mathbf{X}}_i}+c_1)(2\sigma_{\mathbf{X}_i\hat{\mathbf{X}}_i}+c_2)}{(\mu_{\mathbf{X}_i}^2+\mu_{\hat{\mathbf{X}}_i}^2+c_1)(\sigma_{\mathbf{X}_i}^2+\sigma_{\hat{\mathbf{X}}_i}^2+c_2)}, \quad (30)$$

where $\mu_{\mathbf{X}_i}$ and $\mu_{\hat{\mathbf{X}}_i}$ are the averages of $\mathbf{X}_i$ and $\hat{\mathbf{X}}_i$, respectively. $\sigma_{\mathbf{X}_i}^2$, $\sigma_{\hat{\mathbf{X}}_i}^2$ are variances and $\sigma_{\mathbf{X}_i\hat{\mathbf{X}}_i}$ is the covariance of $\mathbf{X}_i$ and $\hat{\mathbf{X}}_i$. c_1 and c_2 are constants.

3) Method Noise: Method noise equals to the difference between the noisy image and the denoised image. The less structural information in the method noise indicates the better quality of the denoised image.

$$M_{noise}(\mathbf{Y},\hat{\mathbf{X}}) = \mathbf{Y} - \hat{\mathbf{X}}. \quad (31)$$

B. Evaluation of NLAM, UD-NLAM and MUD-NLAM

In this subsection, we compare NLAM, UD-NLAM and MUD-NLAM with seven NLM-based methods, i.e. NLM [11], UINTA [14], NLEM [21], INLEM [22], NLPR [23], LJS-NLM [25] and PNLM [24]. We set the parameters of NLAM as $p=2$, $s=7$, $h=20\sigma^2$, those of UD-NLAM as $p=2$, $s=7$, $h=7.5\sigma^2$, $h_s=\sigma$, $\varepsilon=0.16$, and those of MUD-NLAM as $\mathbf{K}=\{1,2,3\}$, $s=7$, $h=0.3\cdot||\mathbf{P}||\sigma^2$, $h_s=\sigma$, $\varepsilon=0.16$. As we use the noise variance σ^2 to compute the unbiased distance, we roughly estimate its value before denoising. For other competing methods, we set their parameters as recommended in related references.

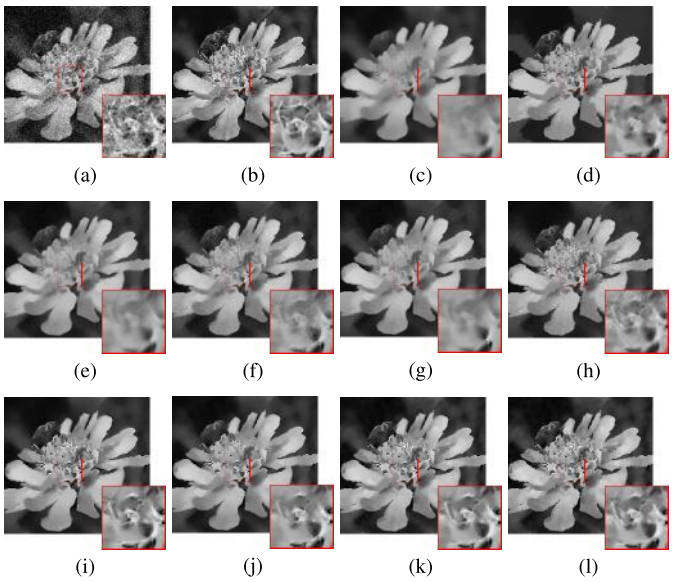

Fig. 5. Denoised results of the *flower* image with noise $\sigma=30$: (a) noisy image (18.60dB); (b) clean image; (c) NLM (24.65dB); (d) UINTA (26.27dB); (e) NLEM (25.41dB); (f) INLEM (26.08dB); (g) NLPR (25.38dB); (h) LJS-NLM (26.84dB); (i) PNLM (27.43dB); (j) NLAM (27.18dB); (k) UD-NLAM (28.04dB); (l) MUD-NLAM (**28.29dB**).

Fig. 4 shows the average PSNR and SSIM values on forty test images of size 256×256. As can be seen, MUD-NLAM performs the best among the existing NLM-based methods on both PSNR and SSIM. From the PSNR results in Fig. 4(a), NLAM is about $1.0dB$ higher than NLM on all situations, and surpasses LJS-NLM by about $0.4dB$ when $\sigma > 50$. By introducing the coupled unbiased distance, UD-NLAM achieves great improvements on PSNR. It improves NLM and PNLM for about $1.8dB$ and $0.4dB$, respectively. UD-NLAM shows slightly worse performance than LJS-NLM when $\sigma<20$ due to the limitation of using single patch. However, UD-NLAM outperforms other existing NLM-based methods when $\sigma\geq20$. By applying multipatch to UD-NLAM, MUD-NLAM further improves the denoising performance, especially when $\sigma < 70$. MUD-NLAM surpasses PNLM by $0.6dB$ approximately. Generally speaking, the denoising performance of all methods decreases as the noise level increases. In terms of the SSIM results in Fig. 4(b), NLAM and UD-NLAM improves NLM by about 8% and 13%, respectively. Besides, UD-NLAM and MUD-NLAM are 3% and 4% higher than PNLM when $\sigma > 40$, respectively. Moreover, all competing denoisng methods drop below 55% when $\sigma\geq90$. However, the denoising performance of UD-NLAM and MUD-NLAM descends more gradually than other methods and reaches 55% when $\sigma=100$.

Fig. 5 shows the denoised results of the *flower* image at noise level $\sigma=30$. As can be observed from the zoom-in part,

TABLE III

PSNR AND SSIM RESULTS OF MUD-NLAM-WS AND OTHER METHODS AT NOISE LEVELS $\sigma = 10, 20, \cdots, 100$.

Image	Method\σ	PSNR(dB) 10	20	30	40	50	60	70	80	90	100	SSIM(%) 10	20	30	40	50	60	70	80	90	100
lena	**MUD-NLAM-WS**	33.38	29.94	28.06	**26.81**	**25.64**	**25.11**	**24.47**	**23.62**	**23.33**	**22.98**	91.90	86.72	82.61	**79.30**	**75.50**	**73.53**	**71.28**	**68.69**	**67.29**	**66.07**
	MUD-NLAM	32.82	29.36	27.52	26.26	25.13	24.36	23.64	22.68	22.30	21.87	91.32	85.38	80.57	76.52	72.86	69.68	66.90	63.18	61.10	59.13
	UD-NLAM	32.72	29.31	27.44	26.08	24.91	24.16	23.55	22.65	22.29	21.90	91.20	85.35	80.73	76.39	72.39	68.99	66.25	62.42	60.22	58.34
	NLAM	32.47	28.55	26.66	25.22	24.13	23.33	22.70	21.98	21.64	21.32	90.86	82.76	77.92	73.52	69.29	65.66	62.83	59.35	57.15	55.08
	BM3D [16]	**33.56**	30.03	28.04	26.49	25.48	24.99	**24.42**	**23.61**	**23.35**	**22.96**	92.36	86.97	**82.62**	78.33	75.14	**73.13**	**70.57**	**68.39**	**66.40**	**65.37**
	DDID [6]	33.51	**30.11**	**28.11**	26.76	25.60	**25.02**	24.34	23.47	23.14	22.80	**92.41**	**87.26**	82.55	78.62	74.56	72.30	69.58	66.56	63.77	63.15
	NL-Bayes [18]	**33.68**	**30.20**	**28.25**	**26.89**	**25.67**	25.01	24.32	23.38	22.91	22.50	**92.56**	**87.19**	**83.03**	**79.38**	**75.52**	72.47	69.57	66.73	63.77	62.36
	PNLM [24]	32.16	28.76	27.10	25.90	24.61	23.74	22.89	21.97	21.58	21.15	91.03	84.83	79.71	75.26	70.49	66.42	62.90	58.85	56.02	53.96
	LJS-NLM [25]	32.59	28.66	26.52	24.89	23.62	22.67	21.74	21.00	20.54	20.13	91.10	83.51	77.46	71.90	66.01	60.90	55.96	51.41	47.57	44.97
	NLPR [23]	31.02	26.84	24.99	23.95	23.29	22.85	22.39	21.76	21.42	21.03	88.08	79.12	73.06	68.55	64.05	59.90	55.28	50.89	47.07	43.70
	INLEM [22]	31.38	27.73	25.75	24.43	23.55	22.95	22.30	21.57	21.08	20.60	89.49	81.27	73.70	67.19	60.91	55.44	49.74	44.78	40.48	37.09
	NLEM [21]	31.44	27.06	25.05	24.01	23.34	22.90	22.40	21.77	21.42	21.03	88.50	79.29	73.16	68.59	63.93	59.67	54.95	50.51	46.60	43.25
	UINTA [14]	31.61	26.95	26.40	24.86	24.06	23.40	22.71	21.67	21.29	20.79	88.98	79.37	75.66	73.29	67.47	67.72	62.55	61.09	57.46	56.77
	NLM [11]	31.58	26.49	24.23	22.85	22.00	21.38	20.84	20.48	20.27	20.07	87.79	77.42	71.07	66.27	62.45	59.39	56.91	54.93	53.45	52.26
montage	**MUD-NLAM-WS**	35.92	32.84	30.73	29.10	**27.81**	**26.70**	**25.84**	**25.10**	**24.65**	**23.94**	95.23	93.46	91.28	**89.79**	**87.55**	**85.76**	**84.48**	**83.15**	**81.27**	**78.51**
	MUD-NLAM	34.11	31.64	29.62	28.26	27.00	26.01	24.86	23.92	23.29	22.20	91.04	90.03	86.57	84.27	81.71	79.81	78.02	75.62	72.77	68.77
	UD-NLAM	34.12	31.45	29.25	27.72	26.49	25.53	24.64	23.88	23.31	22.32	93.19	92.00	88.77	85.82	82.27	79.46	77.17	74.39	70.81	66.54
	NLAM	34.36	31.00	28.74	27.15	25.83	24.81	23.79	22.91	22.37	21.41	94.13	91.88	88.56	85.54	81.71	78.75	75.54	72.18	68.75	64.65
	BM3D [16]	**37.35**	**33.55**	**31.18**	**29.27**	27.59	26.67	25.80	**25.00**	24.30	**23.82**	**96.97**	**94.32**	**91.38**	**88.34**	**86.02**	**84.13**	**81.98**	**80.44**	**77.27**	**75.00**
	DDID [6]	**37.48**	**33.81**	**31.47**	**29.62**	**28.12**	**26.99**	**26.00**	24.90	**24.36**	23.64	**96.81**	**94.03**	**91.29**	88.29	85.15	83.10	80.72	78.72	76.35	74.10
	NL-Bayes [18]	36.85	32.90	30.40	28.70	27.24	26.17	25.38	24.54	23.98	23.27	96.40	93.44	90.19	87.49	83.21	80.46	77.27	75.50	73.68	69.29
	PNLM [24]	34.84	31.23	28.92	27.29	26.06	24.97	24.00	23.07	22.58	21.53	95.58	91.78	88.39	85.44	82.01	78.69	75.81	72.54	69.60	65.49
	LJS-NLM [25]	34.93	31.31	28.72	26.64	24.88	23.72	22.64	21.65	21.17	20.47	94.72	90.27	85.14	80.15	74.09	69.12	64.01	59.05	54.39	50.04
	NLPR [23]	33.25	29.48	27.02	24.56	22.65	21.83	21.32	20.90	20.62	20.23	93.70	88.14	82.79	76.49	69.43	63.53	58.03	53.28	48.19	43.57
	INLEM [22]	33.36	29.57	27.30	25.48	23.74	22.63	21.79	21.14	20.69	20.11	93.68	86.89	79.62	71.92	63.49	56.46	49.82	44.58	39.57	35.01
	NLEM [21]	33.96	29.88	27.18	24.94	22.92	22.08	21.51	21.05	20.76	20.32	94.12	88.34	82.72	76.39	69.25	63.29	57.67	52.81	47.76	43.05
	UINTA [14]	34.56	30.86	28.73	26.94	25.74	24.86	23.81	22.92	22.38	21.68	94.59	90.57	85.36	84.22	79.39	78.92	72.04	73.50	71.06	67.30
	NLM [11]	34.34	29.64	26.60	24.10	22.21	21.39	20.74	20.21	19.98	19.70	94.41	89.55	85.55	81.47	77.07	74.26	71.39	68.94	66.83	64.62
peppers	**MUD-NLAM-WS**	34.30	**31.30**	29.27	**27.68**	**26.86**	**26.06**	**25.15**	**24.48**	**23.74**	**23.20**	93.00	**89.72**	**86.72**	**83.97**	**81.67**	**79.62**	**77.63**	**75.87**	**73.74**	**71.59**
	MUD-NLAM	34.09	30.83	28.80	27.06	26.05	25.01	24.02	23.17	22.39	21.75	92.90	88.93	85.47	81.52	78.44	75.57	72.96	70.15	67.64	64.19
	UD-NLAM	33.58	30.58	28.54	26.77	25.83	24.88	23.99	23.20	22.41	21.83	91.97	88.34	85.12	81.21	78.14	75.21	72.42	69.59	66.74	63.27
	NLAM	33.22	29.39	27.36	25.73	24.71	23.77	22.86	22.14	21.52	21.01	91.62	85.63	82.03	78.15	74.89	71.82	68.66	65.63	63.02	59.57
	BM3D [16]	**34.69**	31.30	**29.29**	27.58	**26.63**	**25.85**	**25.03**	**24.44**	**23.69**	**23.29**	**93.74**	**89.80**	**86.64**	**82.97**	**80.41**	**77.96**	**75.77**	**73.93**	**71.67**	**69.48**
	DDID [6]	34.60	**31.35**	**29.31**	**27.64**	26.59	25.71	24.70	24.17	23.36	22.81	93.49	89.35	86.32	82.84	79.40	76.90	74.55	73.20	70.50	67.56
	NL-Bayes [18]	**34.76**	31.26	29.17	27.52	26.39	25.55	24.58	23.75	22.88	22.39	**93.75**	89.39	86.02	82.33	78.92	76.20	73.29	71.64	68.56	65.79
	PNLM [24]	32.74	29.79	27.94	26.32	25.09	24.07	23.06	22.22	21.62	20.94	91.88	87.35	83.28	79.00	74.98	71.66	68.33	64.82	62.19	58.39
	LJS-NLM [25]	33.25	29.51	27.01	25.07	23.68	22.46	21.59	20.89	20.26	19.64	92.01	86.19	80.69	75.03	69.51	64.41	59.78	55.44	51.18	47.14
	NLPR [23]	31.22	27.81	25.59	23.98	23.09	22.18	21.63	21.17	20.73	20.38	89.50	83.46	77.96	72.06	67.34	62.25	57.71	53.64	49.00	45.00
	INLEM [22]	31.56	28.25	26.14	24.49	23.49	22.53	21.81	21.23	20.65	20.15	90.43	83.90	77.21	69.89	63.45	57.34	51.86	47.21	42.32	38.16
	NLEM [21]	32.01	28.18	25.70	24.03	23.13	22.29	21.72	21.27	20.80	20.42	90.41	83.89	78.02	72.09	67.19	62.07	57.46	53.28	48.62	44.54
	UINTA [14]	32.38	29.17	27.45	25.96	24.40	23.38	22.46	21.82	21.06	20.55	90.57	85.37	79.72	77.69	75.49	73.16	70.42	66.94	63.37	60.98
	NLM [11]	32.29	27.66	24.84	22.91	21.75	20.87	20.41	20.04	19.72	19.39	90.17	82.81	76.84	71.84	67.71	64.19	61.82	59.39	57.73	55.36

NLM, UINTA, INLEM and NLPR over-smooth the image and lose most details of the flower, while NLAM preserves more detail information. LJS-NLM and PNLM keep some structures and details but there is obvious noise left around the edges of the flower. Compared with these competing NLM-based methods, UD-NLAM and MUD-NLAM perform better in both reducing noise and in preserving details.

In summary, NLAM outperforms some of the NLM-based methods. UD-NLAM and MUD-NLAM further improve the denoising performance of NLAM, and achieve much better performance than the existing NLM-based methods in both quantitative and visual evaluations. Besides, in comparison of NLM, NLAM and UD-NLAM, we can conclude that the improved denoising performance of the proposed methods is caused by both adaptive weight ω_{ij} and the proposed distance measure.

C. Evaluation of MUD-NLAM-WS

In order to show the denoising performance of MUD-NLAM-WS, we compare it with several advanced denoising methods, including NL-Bayes [18], DDID [6], BM3D [16], Slanted Butterworth function based NLM (NLM-SB) [35], Meyer [36], NLM-SB2 [35], SB-Meyer [35] and seven NLM-based methods in the previous subsection. NLM-SB [35] improves NLM by constructing the low-rank approximation of an NLM operator. Meyer [36] denoises an image by using the eigenvectors of the patch-graph Laplacian. NLM-SB2 and SB-Meyer [35] are two improvement methods of NLM-SB by the simple two-stage scheme and Meyer's two-stage scheme, respectively.

TABLE IV

PARAMETER SETTINGS

Image size	Parameter	Value: Guide image generation	Image denoising: Iteration 1	Image denoising: Iteration 2
512×512	**K**	{1,2,3}	{1,2}	
	s	11	12	
	h	$0.25 \cdot \|\|\mathbf{P}\|\| \sigma^2$	$30\sigma^2$	$6\sigma^2$
	h_s	$0.7 \cdot \sigma$	98	162
	γ	-	0.5-0.002σ	0.5
	ε	0.16	0.25	
256×256	**K**	{1,2,3}	{1,2}	
	s	7	12	
	h	$0.3 \cdot \|\|\mathbf{P}\|\| \sigma^2$	$45\sigma^2$	$4\sigma^2$
	h_s	σ	98	
	γ	-	0.5-0.002σ	0.5
	ε	0.16	0.25	
120×120	**K**	{1,2,3}	{1,2}	
	s	9	17	15
	h	$0.25 \cdot \|\|\mathbf{P}\|\| \sigma^2$	$30\sigma^2$	$6\sigma^2$
	h_s	$0.7 \cdot \sigma$	162	98
	γ	-	0.5-0.002σ	0.5
	ε	0.16	0.25	

According to different image sizes, the experiments can be divided into three groups below:

- **Group 1:** We compare the proposed NLAM, UD-NLAM, MUD-NLAM and MUD-NLAM-WS with NL-Bayes, DDID, BM3D and seven competing methods in Section VII-B. The experiments are implemented on a set of test images with size of about 256×256. The noise level σ ranges from 10 to 100.

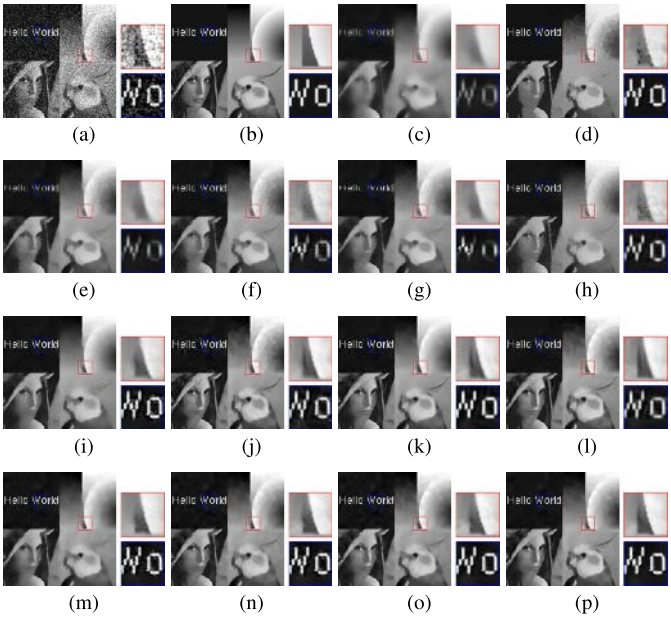

Fig. 6. Denoised results of the *montage* image with noise $\sigma = 50$: (a) noisy image; (b) clean image; (c) NLM; (d) UINTA; (e) NLEM; (f) INLEM; (g) NLPR; (h) LJS-NLM; (i) PNLM; (j) NL-Bayes; (k) DDID; (l) BM3D; (m) NLAM; (n) UD-NLAM; (o) MUD-NLAM; (p) MUD-NLAM-WS.

Fig. 7. Denoised results of the *cameraman* image with noise $\sigma = 70$: (a) noisy image; (b) clean image; (c) NLM; (d) UINTA; (e) NLEM; (f) INLEM; (g) NLPR; (h) LJS-NLM; (i) PNLM; (j) NL-Bayes; (k) DDID; (l) BM3D; (m) NLAM; (n) UD-NLAM; (o) MUD-NLAM; (p) MUD-NLAM-WS.

- **Group 2:** We conduct the comparative experiments on MUD-NLAM, MUD-NLAM-WS, NL-Bayes, DDID and BM3D and NLM-SB. Different from **Group 1**, we use signal-to-noise ratio (SNR) to measure the noise level in this group. The simulations are implemented on a set of images with size of about 512×512 at $SNR = 0.5, 0.75$ and 1, respectively.
- **Group 3:** We compare MUD-NLAM and MUD-NLAM-WS with NLM-SB, NLM-SB2, SB-Meyer, Meyer, NL-Bayes, DDID and BM3D. The experiments are implemented on a set of test images with size 120×120 at $SNR = 1$.

In each group of experiments, the parameter settings of MUD-NLAM-WS are shown in Table IV. For the competing methods, we set their parameters as recommended in corresponding references. Related results are described as follows:

1) Group 1: In Table III, the best two results are marked with bold font in blue and black, respectively. By analysing these quantitative results, we find that MUD-NLAM-WS surpasses PNLM by about $1.5dB$ at each noise level in terms of PSNR. In view of SSIM, MUD-NLAM-WS performs the best when $\sigma > 40$. Furthermore, MUD-NLAM-WS achieves competitive results with BM3D especially when σ varies from 40 to 100. This is because, when noise level σ is small, BM3D takes advantage of the enhanced sparse representation in transform domain to effectively remove noise and retain details. However, as noise level increases, the sparse representation also makes BM3D more sensitive to the noise patterns and produces noticeable artifacts.

Fig. 6 presents the denoised results of the *montage* image with noise $\sigma = 50$. As can be observed in the zoom-in part within red square of the clean image (Fig. 6(b)), there is obvious sawtooth edge on the contour of the cropped circle. UD-NLAM, MUD-NLAM and MUD-NLAM-WS reconstruct this detail in their denoised results. However, the sawtooth textures are almost disappeared in the denoised images by the other methods. For the letters shown in the zoom-in part highlighted within blue square, UD-NLAM, MUD-NLAM and MUD-NLAM-WS outperform other NLM-based methods in noise removal and edge preservation. In addition, the competing denoising methods introduce some ringing artifacts to the edges of letters or on the background in the denoised results. However, MUD-NLAM-WS generates significantly less artifacts and retains much more detail structures than other methods.

Fig. 7 shows the denoised results of the *cameraman* image with noise $\sigma = 70$. As shown in the zoom-in part, NLM, NLEM, INLEM and NLPR over-smooth the image. As a result, it is hard to identify the structures and details of the camera. NLEM, INLEM and NLPR also bring many ringing artifacts to the background of the denoised images. LJS-NLM, PNLM and NL-Bayes leave some noise around the edges of the camera. UINTA produces clear edges, but loses some structure information. UD-NLAM and MUD-NLAM perform better in denoising and keeping edges than other NLM-based methods. MUD-NLAM-WS preserves more details and produces less artifacts than BM3D and DDID.

Fig. 8 shows the corresponding method noises of the denoised results in Fig. 7. Observing the denoised results of

TABLE V

PSNR COMPARISON AMONG MUD-NLAM-WS, MUD-NLAM AND OTHER FOUR METHODS AT SNR = 0.5, 0.75 AND 1

Noise level	SNR=0.5						SNR=0.75						SNR=1					
Image	NLM-SB [35]	NL-Bayes [18]	DDID [6]	BM3D [16]	**MUD-NLAM**	**MUD-NLAM -WS**	NLM-SB [35]	NL-Bayes [18]	DDID [6]	BM3D [16]	**MUD-NLAM**	**MUD-NLAM -WS**	NLM-SB [35]	NL-Bayes [18]	DDID [6]	BM3D [16]	**MUD-NLAM**	**MUD-NLAM -WS**
crowd	20.36	22.01	22.08	**22.39**	21.27	22.23	22.30	24.02	23.94	**24.19**	23.26	24.04	23.67	25.44	25.30	25.42	24.78	**25.44**
flower	21.27	23.85	24.15	24.36	23.51	**24.48**	22.14	26.20	26.23	26.43	25.49	**26.47**	25.10	27.62	27.83	27.87	26.90	**27.94**
flowers	20.33	21.68	21.77	**22.00**	21.14	21.89	22.10	23.59	23.58	**23.70**	22.03	23.63	23.23	24.79	24.79	24.76	24.33	**24.79**
girlface	21.45	24.23	24.74	24.92	23.87	**24.99**	22.58	26.45	26.65	26.80	25.73	**26.80**	25.24	27.71	27.87	**27.94**	26.91	27.86
monarch	21.48	24.72	24.72	24.79	23.78	**25.01**	24.09	26.84	**27.06**	26.76	26.18	27.01	25.57	28.38	**28.73**	28.20	27.51	28.36
pens	20.87	23.40	24.01	23.92	23.21	**24.21**	22.01	25.39	25.69	**25.85**	25.02	25.82	24.29	26.74	27.06	**27.19**	26.18	27.04
sailboat	14.24	20.87	20.99	21.02	20.35	**21.17**	16.38	22.74	22.74	**23.15**	22.11	22.89	19.76	23.99	23.93	**24.22**	23.42	24.04
tulips	19.37	22.27	22.70	22.75	21.95	**23.05**	20.38	24.46	24.67	24.81	23.95	**25.04**	23.10	25.96	26.04	26.09	25.42	**26.32**
zelda	21.77	24.48	25.29	25.01	24.54	**25.50**	24.07	26.62	27.31	27.09	26.42	**27.34**	25.52	28.09	**28.64**	28.39	27.47	28.57

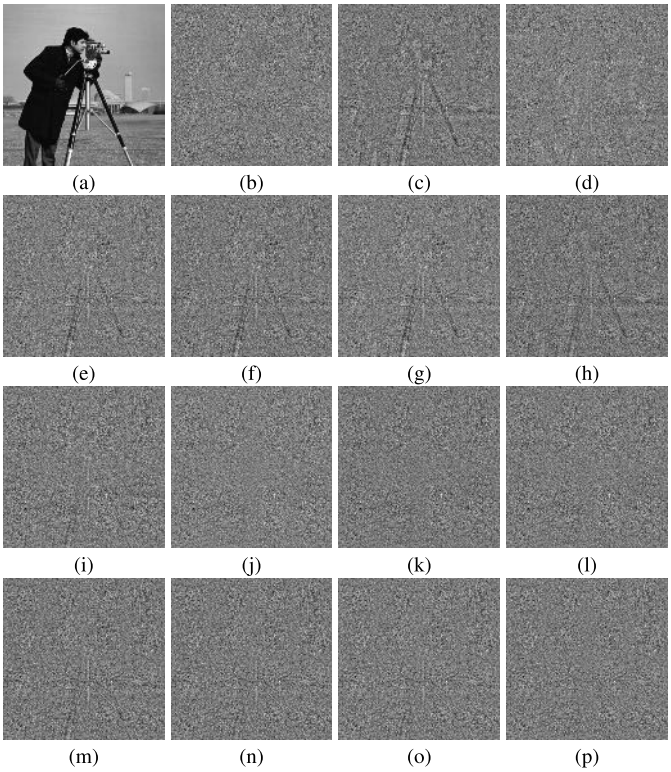

Fig. 8. Method noise of the *cameraman* image with noise $\sigma = 70$: (a) clean image; (b) noise; (c) NLM; (d) UINTA; (e) NLEM; (f) INLEM; (g) NLPR; (h) LJS-NLM; (i) PNLM; (j) NL-Bayes; (k) DDID; (l) BM3D; (m) NLAM; (n) UD-NLAM; (o) MUD-NLAM; (p) MUD-NLAM-WS.

NLM-based methods in Figs. 8(c)-(i) and (m), we can identify the outline of the cameraman. Compared with competing NLM-based methods, the method noises of UD-NLAM and MUD-NLAM show much less structure information of the object. In addition, there is almost no noticeable structural information remaining in the method noises of NL-Bayes, DDID, BM3D and MUD-NLAM-WS. This demonstrates their excellent denoising performance.

2) Group 2: As shown in Table V, MUD-NLAM and MUD-NLAM-WS have better denoising performance than NLM-SB on all situations. MUD-NLAM-WS surpasses NLM-SB, NL-Bayes and DDID by about $2.9dB$, $0.3dB$ and $0.2dB$ in average, respectively. Besides, MUD-NLAM-WS obtains higher PSNR values than BM3D in most cases. The denoised results of NLM-SB are from the reference [35].

TABLE VI

PSNR COMPARISON AMONG MUD-NLAM-WS, MUD-NLAM AND OTHER SEVEN METHODS AT SNR = 1

Image	NLM-SB [35]	NLM-SB2 [35]	SB-Meyer [35]	Meyer [36]	NL-Bayes [18]	DDID [6]	BM3D [16]	**MUD-NLAM**	**MUD-NLAM -WS**
boat	21.44	22.01	22.72	21.33	23.79	23.72	**23.81**	23.36	23.80
couple1	21.51	22.54	23.50	23.06	24.29	24.34	24.34	23.86	**24.41**
couple2	21.81	23.17	24.46	24.61	25.30	25.53	**25.58**	24.79	25.53
lena	21.75	22.39	23.12	22.44	24.41	24.43	24.48	23.76	**24.51**
man1	21.45	21.70	22.04	21.16	24.82	24.64	24.60	24.36	**24.97**
man2	21.88	21.47	22.08	20.01	26.24	26.11	26.25	25.75	**26.33**
woman blonde	21.60	21.89	23.18	23.64	26.27	26.29	25.97	25.40	**26.31**

Fig. 9 shows the visual results of NL-Bayes, DDID, BM3D, MUD-NLAM and MUD-NLAM-WS on the *girlface* image at $SNR = 1$. As can be observed, NL-Bayes, DDID and BM3D yield many noticeable ringing artifacts in the face, while MUD-NLAM-WS removes the noise more effectively than the competing methods.

3) Group 3: As can be seen in Table VI, MUD-NLAM-WS and MUD-NLAM perform better than NLM-SB, NLM-SB2, SB-Meyer and Meyer in terms of PSNR on all test images. Besides, MUD-NLAM-WS achieves comparable performance with BM3D, DDID and NL-Bayes, especially on the *boat* and *lena* images. However, on the *man*1 and *woman blonde* images, the PSNR values of MUD-NLAM-WS are greater than that of BM3D for about $0.3dB$. The denoised results of NLM-SB, NLM-SB2, SB-Meyer and Meyer are from the reference [35].

From the results of the three groups of experiments, the main conclusions can be drawn as follows:

- MUD-NLAM-WS produces much more pleasant denoised images while achieving higher PSNR and SSIM values than the competing NLM-based methods.
- Although PSNR results of MUD-NLAM-WS over BM3D are not significant, MUD-NLAM-WS improves BM3D up to 4% in terms of SSIM. In addition, MUD-NLAM-WS generates more visually pleasant denoised results than BM3D. As can be seen in Figs. 6, 7 and 9, BM3D yields noticeable ringing artifacts and false contours in smooth areas, especially under the cases with a high level of noise. These artifacts seriously degrade the visual quality of denoised images of BM3D. On the other hand, MUD-NLAM-WS produces much less artifacts while effectively preserving structure and detail information.
- We have conducted the experiments on the images with two different resolutions, e.g. 256×256 and 512×512.

(a) (b) (c) (d) (e)

Fig. 9. Denoised results of the $girlface$ image with $SNR = 1$: (a) clean image; (b) NL-Bayes; (c) DDID; (d) BM3D; (e) MUD-NLAM-WS.

The denoised results of 256×256 are shown in Table III and Figs. 6-8, and the denoised results of 512×512 are shown in Tables V and Fig. 9. As can be seen, MUD-NLAM-WS produces more appealing denoised results than other competing denoising methods in terms of quantitative evaluations and visual comparison. Therefore, with appropriate parameter settings, the image resolution does not influence the denoising performance of our MUD-NLAM-WS.

VIII. Conclusion

In this paper, we introduced the non-local adaptive means (NLAM) for image denoising. The weight ω_{ij} is considered as an optimization variable that is adaptive to the image content. By measuring the image patch similarity, NLAM updates weight ω_{ij} in each denoising iteration. We then defined three unbiased distances, i.e. pixel-pixel unbiased distance, patch-patch unbiased distance and coupled unbiased distance. Since these distances contain the structure information of noisy image and the denoised image in each iteration, they provide more robustness to measure the image pixel/patch similarity than the traditional Euclidean distance. Using the coupled unbiased distance, we proposed the unbiased distance non-local adaptive means (UD-NLAM), and then extended UD-NLAM into its multipatch version to adapt different noise levels and proposed multipatch UD-NLAM (MUD-NLAM). Integrating wavelet shrinkage, we further introduced a novel denoising method, named MUD-NLAM with wavelet shrinkage (MUD-NLAM-WS). We conducted extensive comparison experiments between our methods and several existing denoising methods. The quantitative evaluations and visual comparisons showed that NLAM outperforms some of the existing NLM-based methods. UD-NLAM and MUD-NLAM further improve the denoising performance of NLAM, and MUD-NLAM-WS is superior to the competing state-of-the-art denoising methods in both noise removal and detail preservation.

Since the proposed methods focus on grayscale image denoising, our future work will investigate how to extend the proposed methods to color image denoising. For example, we will introduce a novel distance criteria to measure the similarity between two color image pixels/patches. Moreover, integrating non-local means into deep learning framework is interesting to explored.

Acknowledgment

The authors would like to sincerely thank the editors and anonymous reviewers for their valuable comments and suggestions that helped to significantly improve the quality of our manuscript.

References

[1] L. P. Yaroslavsky, *Digital Picture Processing—An Introduction*. Berlin, Germany: Springer-Verlag, 1985.
[2] S. M. Smith and J. M. Brady, "SUSAN—A new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, 1997.
[3] J.-S. Lee, "Digital image smoothing and the sigma filter," *Comput. Vis., Graph. Image Process.*, vol. 24, no. 2, pp. 255–269, 1983.
[4] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.
[5] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.
[6] C. Knaus and M. Zwicker, "Dual-domain image denoising," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 440–444.
[7] C. Knaus and M. Zwicker, "Dual-domain filtering," *SIAM J. Image Sci.*, vol. 8, no. 3, pp. 1396–1420, 2015.
[8] C. Knaus and M. Zwicker, "Progressive image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3114–3125, Jul. 2014.
[9] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.
[10] D. K. Hammond and E. P. Simoncelli, "Image modeling and denoising with orientation-adapted Gaussian scale mixtures," *IEEE Trans. Image Process.*, vol. 17, no. 11, pp. 2089–2101, Nov. 2008.
[11] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. San Diego, CA, USA, Jun. 2005, pp. 60–65.
[12] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.
[13] S. P. Awate and R. T. Whitaker, "Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2005, pp. 44–51.
[14] S. P. Awate and R. T. Whitaker, "Unsupervised, information-theoretic, adaptive image filtering for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 364–376, Mar. 2006.
[15] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.
[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] M. Lebrun, "An analysis and implementation of the BM3D image denoising method," *Image Process. On Line*, vol. 2, pp. 175–213, Aug. 2012.
[18] M. Lebrun, A. Buades, and J.-M. Morel, "Implementation of the 'non-local Bayes' (NL-Bayes) image denoising algorithm," *Image Process. Line*, vol. 3, pp. 1–42, Jun. 2013.
[19] M. Lebrun, A. Buades, and J.-M. Morel, "A nonlocal Bayesian image denoising algorithm," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1665–1688, 2013.

[20] T. Dai, C.-B. Song, J.-P. Zhang, and S.-T. Xia, "PMPA: A patch-based multiscale products algorithm for image denoising," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 4406–4410.
[21] K. N. Chaudhury and A. Singer, "Non-local Euclidean medians," *IEEE Signal Process. Lett.*, vol. 19, no. 11, pp. 745–748, Nov. 2012.
[22] Z. Sun and S. Chen, "Analysis of non-local Euclidean medians and its improvement," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 303–306, Apr. 2013.
[23] K. N. Chaudhury and A. Singer, "Non-local patch regression: Robust image denoising in patch space," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2013, pp. 1345–1349.
[24] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, "Probabilistic non-local means," *IEEE Signal Process. Lett.*, vol. 20, no. 8, pp. 763–766, Aug. 2013.
[25] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, "James–Stein type center pixel weights for non-local means image denoising," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 411–414, Apr. 2013.
[26] V. Fedorov and C. Ballester, "Affine non-local means image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2137–2148, May 2017.
[27] R. Lan, Y. Zhou, Y. Y. Tang, and C. L. P. Chen, "Image denoising using non-local fuzzy means," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Jul. 2015, pp. 196–200.
[28] X. Li, Y. Zhou, J. Zhang, and L. Wang, "Unbiased distance based non-local fuzzy means," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Apr. 2018, pp. 1423–1427.
[29] G. Facciolo, P. Arias, V. Caselles, and G. Sapiro, "Exemplar-based interpolation of sparsely sampled images," in *Proc. 7th Int. Conf. Energy Minimization Methods Comput. Vis. Pattern Recognit. (EMMCVPR)*. Berlin, Germany: Springer-Verlag, 2009, pp. 331–344.
[30] P. Arias, V. Caselles, G. Facciolo, V. Lazcano, and R. Sadek, "Nonlocal variational models for inpainting and interpolation," *Math. Models Methods Appl. Sci.*, vol. 22, no. 2, Aug. 2012, Art. no. 1230003.
[31] E. Luo, S. Pan, and T. Nguyen, "Generalized non-local means for iterative denoising," in *Proc. 20th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2012, pp. 260–264.
[32] H. Yu, L. Zhao, and H. Wang, "Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2364–2369, Oct. 2009.
[33] J. B. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-25, no. 3, pp. 235–238, Jun. 1977.
[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[35] V. May, Y. Keller, N. Sharon, and Y. Shkolnisky, "An algorithm for improving non-local means operators via low-rank approximation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1340–1353, Mar. 2016.
[36] F. G. Meyer and X. Shen, "Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising," *Appl. Comput. Harmon. Anal.*, vol. 36, no. 2, pp. 326–334, 2014.

Xiaoyao Li received the B.S. degree in mathematics and applied mathematics from the China University of Petroleum, Beijing, China, in 2012.

She is currently pursuing the Ph.D. degree with Control Science and Engineering, College of Electrical and Information Engineering, Hunan University, Changsha, China. She was a Research Assistant with the Department of Computer and Information Science, University of Macau in 2017. Her current research interests include image denoising and signal/image processing.

Yicong Zhou (M'07-SM'14) received the B.S. degree from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees from Tufts University, Massachusetts, USA, all in electrical engineering.

He is an Associate Professor and the Director of the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau. His research interests include image processing and understanding, computer vision, machine learning, and multimedia security.

Dr. Zhou serves as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and several other journals. He is the Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. Dr. Zhou is a Senior Member of the International Society for Optical Engineering (SPIE). He was a recipient of the Third Price of Macau Natural Science Award in 2014.

Jing Zhang received the B.S., M.S., and Ph.D. degrees from Hunan University, Changsha, China, in 1982, 1984, and 1997, respectively.

He is currently a Professor with the College of Electrical and Information Engineering, Hunan University, Changsha. He has published over 300 papers in journals and conferences. His research interests include optimal control, fuzzy control, intelligent control for complex industrial system, and pattern recognition.

Dr. Zhang was a recipient of the Second Prize and the First Prize of National Scientific and Technological Progress Awards of China in 2010 and 2018, respectively.

Lianhong Wang received the B.S., M.S., and Ph.D. degrees from Hunan University, Changsha, China, in 1993, 2002, and 2009, respectively.

She is currently an Associate Professor with the College of Electrical and Information Engineering, Hunan University, Changsha. She was a visiting scholar at Brandeis University from 2011 to 2012. She has published over 30 papers in journals and conferences. Her research interests include signal/image processing, data mining, and modern network communication technology.

Dr. Wang was a recipient of the First Prize of Teaching Achievement Award for Higher Education in Hunan Province.