UNBIASED DISTANCE BASED NON-LOCAL FUZZY MEANS

Xiaoyao Li[1,2]*, Yicong Zhou*[2]*, Jing Zhang*[1]*, Lianhong Wang*[1]

[1]College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
[2]Department of Computer and Information Science, University of Macau, Macau 999078, China
lxyimayday@gmail.com, yicongzhou@umac.mo

ABSTRACT

This paper first introduces several unbiased distances to measure the similarity between image pixels and between image patches. Based on these unbiased distances, we propose a new non-local means denoising method, named Unbiased Distance based Non-local Fuzzy Means (UDNLFM). UDNLFM considers the weight as a fuzzy variable and updates its value in each denoising iteration. Experiments and comparisons demonstrate that UDNLFM outperforms state-of-the-art non-local means methods for image denoising.

***Index Terms*—** Image denoising, non-local means, non-local fuzzy means, unbiased distance

1. INTRODUCTION

Since noise is unavoidable in image acquisition and transmission, image denoising becomes an essential step before further processing. It aims at removing the noise effectively and also performing well in detail preservation.

Recently, numerous denoising methods have been advanced to reduce Gaussian noise. Examples include block-matching and 3D filtering [1], non-local Bayes [2], principal component analysis with local pixel grouping [3], non-local dual image denoising [4], progressive image denoising [5] and patch-based multiscale products algortithm [6]. Among denoising methods, Non-local Means (NLM) [7–9] was first introduced by Buades et al. and it attracted lots of attention. NLM uses image patch as a unit and is more robust in image denoising, because image patch contains more structure information of an image than pixels.

The principle of NLM is to calculate the weighted average of all neighbour patches in a given search window. The weight measures the similarity between two patches. Given a noisy image as $\mathbf{Y}$, it can be expressed as a linear combination of a clean image $\mathbf{X}$ and a noise model $\mathbf{N}$

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}. \tag{1}$$

This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/016/2015/A1 and by the Research Committee at University of Macau under Grant MYRG2016-00123-FST.

Firstly, NLM calculates the weight ω_{ij} for each pixel in a search window $\mathbf{S}_i$,

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

where $j \in \mathbf{S}_i$. $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are noisy patches located at i and j of $\mathbf{Y}$. h is the smoothing parameter. NLM then obtains the target patch $\hat{\mathbf{X}}_i$ as

$$\hat{\mathbf{X}}_i = \frac{\sum\limits_{j \in \mathbf{S}_i} \omega_{ij} \mathbf{Y}_j}{\sum\limits_{j \in \mathbf{S}_i} \omega_{ij}}. \tag{3}$$

$\hat{\mathbf{X}}_i$ is an image patch centered at i of denoised image $\hat{\mathbf{X}}$.

Due to its robustness in removing Gaussian noise, many NLM-based methods were proposed, such as non-local Euclidean medians (NLEM) [10], improved NLEM (INLEM) [11], Probabilistic NLM (PNLM) [12] and NLM with local James-Stein type center pixel weights (LJS-NLM) [13]. NLM and its improvements consider weight w_{ij} as a constant. That means they only calculate ω_{ij} once and keep it unchanged during later iterative denoising processes. This is improper because the denoised image and patch similarity will change after each iteration. To address this issue, non-local fuzzy means (NLFM) [14] was proposed. NLFM considers weight ω_{ij} as a fuzzy variable and iteratively updates its value along with denoised image after each denoising iteration. However, its denoising performance is not promising due to the fact that Euclidean distance has limitation to measure the similarity between two image patches.

In this paper, we propose three unbiased distances, namely pixel-pixel unbiased distance, patch-patch unbiased distance and combined unbiased distance. They are robust to measure the similarity between image pixels or between patches. Then we propose Unbiased Distance based NLFM (UDNLFM). Similar to NLFM, UDNLFM considers weight ω_{ij} as a variable and updates its value in each denoising iteration via computing the combined unbiased distances between patches. Experiments have shown that UDNLFM is robust to remove Gaussian noise and superior to several NLM-based methods with respect to quantitative measure and visual quality of denoised image.

2. PROPOSED METHOD

After reviewing the optimization models of several existing denoising methods, this section proposes three unbiased distances to measure the similarity between pixels or pathces, then presents the proposed Unbiased Distance based NLFM (UDNLFM) for image denoising.

2.1. Optimization Models

As NLM aims at finding the optimal target patch in a given search window, we can transform this process into an optimization problem [11]. The optimization models of several denoising methods are listed in Table 1. The related NLM-based methods are traditional NLM, NLEM, INLEM, PNLM and NLFM. Table 1 also lists the optimazation models of denoising filters, such as mean filter, Gaussian filter and median filter. In this table, $\hat{\mathbf{X}}(i)$, $\mathbf{X}(i)$ and $\mathbf{Y}(j)$ are the i/jth pixel of denoised image (or target image) $\hat{\mathbf{X}}$, clean image $\mathbf{X}$ and noisy image $\mathbf{Y}$, respectively. $\hat{\mathbf{X}}_i$, $\mathbf{X}_i$ and $\mathbf{Y}_j$ are patches centered at i/j in $\hat{\mathbf{X}}$, $\mathbf{X}$ and $\mathbf{Y}$, repectively.

Table 1. 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	$\hat{\mathbf{X}}(i) = \arg\min_{\mathbf{X}(i)} \sum_{j\in\mathbf{S}_i} \omega_{ij} (\mathbf{X}(i) - \mathbf{Y}(j))^2$
Median Filter	$\hat{\mathbf{X}}(i) = \arg\min_{\mathbf{X}(i)} \sum_{j\in\mathbf{S}_i} \lvert\mathbf{X}(i) - \mathbf{Y}(j)\rvert$
NLM [11]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} \omega_{ij} \lvert\lvert\mathbf{X}_i - \mathbf{Y}_j\rvert\rvert^2$
NLEM [10]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} \omega_{ij} \lvert\lvert\mathbf{X}_i - \mathbf{Y}_j\rvert\rvert$
INLEM [11]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} \sqrt{\omega_{ij}} \lvert\lvert\mathbf{X}_i - \mathbf{Y}_j\rvert\rvert$
PNLM [12]	$\hat{\mathbf{X}}_i = \arg\min_{\mathbf{X}_i} \sum_{j\in\mathbf{S}_i} f_{ij} \lvert\lvert\mathbf{X}_i - \mathbf{Y}_j\rvert\rvert^2$
NLFM [14]	$\{\hat{\mathbf{X}}_i, \hat{\omega}_{ij}\} = \arg\min_{\mathbf{X}_i,\omega_{ij}} \sum_{j\in\mathbf{S}_i} \omega_{ij}^m \lvert\lvert\mathbf{X}_i - \mathbf{Y}_j\rvert\rvert^2$

Mean filter denoises an image through averaging all pixels in a search window, and assigns the resulting value to center pixel. Gaussian filter is a kind of weighted average filters. Its weight is a given Gaussian kernel. Median filter is used to find the median value in a search window. Different from these denoising filters, NLM is a patch-based denoising method and its weight is calculated from patch similarity. Based on NLM, NLEM improves the denoising effect by replacing $||\mathbf{X}_i - \mathbf{Y}_j||^2$ with $||\mathbf{X}_i - \mathbf{Y}_j||$. For further improvement, INEM changes the weight of NLEM as $\sqrt{\omega_{ij}}$. Unlike these NLM-based methods, PNLM applies a new probabilistic weight f_{ij} that can better reflects the patch similarity.

As can be seen, these denoising filters and NLM-based methods calculate weight ω_{ij} only once and keep it unchanged in later iterations. This sometimes leads to a worse denoising performance because the similarity between patches changes in each denoising iteration. In order to overcome this weakness, NLFM considers ω_{ij} as a fuzzy variable and updates both denoised image $\hat{\mathbf{X}}$ and weight ω_{ij} iteratively. m of ω_{ij} is an exponential parameter and can nonlinearly map the weight into an appropriate value to enhance the denoising effect. However, the denoising performance of NLFM is not convincing as well.

In Section 2.3, we will propose UDNLFM as a novel NLM-based method that considers weight ω_{ij} as a fuzzy variable and defines new unbiased distance to replace the traditional Euclidean distance.

2.2. Unbiased Distances

Firstly, we define a new unbiased distance. Given two noisy pixels $\mathbf{Y}(i)$ and $\mathbf{Y}(j)$, their squared pixel-pixel unbiased distance is

$$\mathbb{D}_{\text{U}}^2\big(\mathbf{Y}(i), \mathbf{Y}(j)\big) = \big(\mathbf{Y}(i) - \mathbf{Y}(j)\big)^2 - 2\sigma^2, \tag{4}$$

where i and j are the pixel locations in noisy image $\mathbf{Y}$. σ^2 is the variance of noise. From the viewpoint of statistics, $\big(\mathbf{Y}(i) - \mathbf{Y}(j)\big)^2 - 2\sigma^2$ is an unbiased estimator of $\big(\mathbf{X}(i) - \mathbf{X}(j)\big)^2$ [7, 9, 15]. Then, we add up all the unbiased pixel distances within a patch window $\mathbf{P}$ to obtain the squared patch-patch unbiased distance as

$$\mathbb{D}_{\text{U}}^2\big(\mathbf{Y}_i, \mathbf{Y}_j\big) = ||\mathbf{Y}_i - \mathbf{Y}_j||^2 - 2||\mathbf{P}||\sigma^2, \tag{5}$$

$||\mathbf{P}||$ is the size of $\mathbf{P}$. Using the same principle, we can get a generalized squared patch-patch unbiased distance between a noisy patch $\mathbf{Y}_j$ and a denoised patch (or target patch) $\hat{\mathbf{X}}_i$ as

$$\begin{aligned}\mathbb{D}_{\text{U}}^2\big(\hat{\mathbf{X}}_i, \mathbf{Y}_j\big) =& ||\hat{\mathbf{X}}_i - \mathbf{Y}_j||^2 \\ &- ||\mathbf{P}||\Big(\sum_{l\in\mathbf{S}_i} \omega_{il}^2 - 2\omega_{ij} + 1\Big)\sigma^2,\end{aligned} \tag{6}$$

where $\sum_{l\in\mathbf{S}_i} \omega_{il} = 1$ and $\hat{\mathbf{X}}_i = \sum_{l\in\mathbf{S}_i} \omega_{il}\mathbf{Y}_l$. If $\omega_{ii} = 1$ while $\omega_{il} = 0$ ($l \in \mathbf{S}_i, l \neq i$), we will get $\hat{\mathbf{X}}_i = \mathbf{Y}_i$ and thus Eq. (6) is reduced to Eq. (5).

Using patch-patch unbiased distance, we introduce a new distance, named the combined unbiased distance. It will be applied in our proposed UDNLFM. The squared combined unbiased distance is defined by

$$\begin{aligned}&\mathbb{D}_{\text{C}}^2\big(\hat{\mathbf{X}}_i, \mathbf{Y}_j\big) = \\ &\alpha \cdot \max\big[0, \mathbb{D}_{\text{U}}^2(\hat{\mathbf{X}}_i, \mathbf{Y}_j)\big] + \beta \cdot \big(\bar{\hat{x}}_i - \bar{\hat{x}}_j\big)^2.\end{aligned} \tag{7}$$

$\bar{\hat{x}}_i$ and $\bar{\hat{x}}_j$ are average pixel values of the related denoised patch. α and β are trade-off parameters.

2.3. UDNLFM

In this subsection, we introduce a novel NLM-based method, named Unbiased Distance based Non-local Fuzzy Means (UDNLFM).

Using this combined unbiased distance, we introduce the optimization model of UDNLFM as

$$\{\hat{\mathbf{X}}_i, \hat{\omega}_{ij}\} = \underset{\mathbf{X}_i, \omega_{ij}}{\arg\min} \sum_{j \in \mathbf{S}_i} \omega_{ij} \mathbb{D}_c^2(\mathbf{X}_i, \mathbf{Y}_j). \tag{8}$$

In order to solve this optimization problem, we initialize $\hat{\mathbf{X}}_i^{(0)} = \mathbf{Y}_i$ and then update ω_{ij} and $\hat{\mathbf{X}}_i$ alternatively using the following two equations.

$$\omega_{ij}^{(t+1)} = \exp\left(-\frac{\mathbb{D}_c^2(\hat{\mathbf{X}}_i^{(t)}, \mathbf{Y}_j)}{h^2}\right) \cdot H_{ij}, \tag{9}$$

$$\hat{\mathbf{X}}_i^{(t+1)} = \frac{\sum_{j \in \mathbf{S}_i} \omega_{ij}^{(t)} \mathbf{Y}_j}{\sum_{j \in \mathbf{S}_i} \omega_{ij}^{(t)}}, \tag{10}$$

where $\hat{\mathbf{X}}_i^{(t)}$ is the denoised image patch in the tth denoising iteration, where $t = 0, 1, 2, \cdots$. After $(t+1)$ iterations, we update the weight and the denoised patch as $\omega_{ij}^{(t+1)}$ and $\hat{\mathbf{X}}_i^{(t+1)}$. The spatial kernel $H_{ij} = \exp\left(-\frac{(i-j)^2}{h_s}\right)$ and h_s is the spatial parameter. Therefore, the new weight ω_{ij} contains both patch similarity and spatial information. The detail procedures of UDNLFM is shown in Algorithm 1.

Algorithm 1 UDNLFM

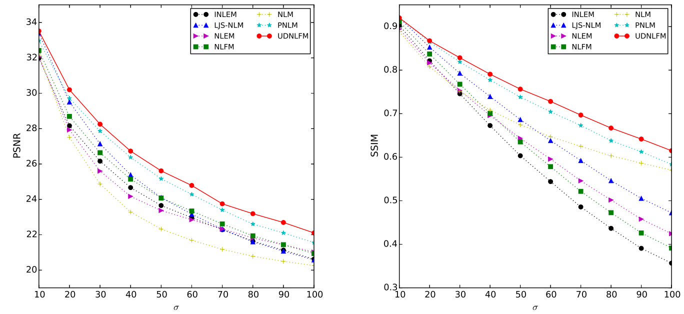
Input: The noisy image $\mathbf{Y}$, the radius of patch k, the radius of search window s and other parameters h, h_s, α, β.

- Step 1: Extract a patch $\mathbf{Y}_i$ with radius k centered at each pixel i in $\mathbf{Y}$.
- Step 2: For each pixel i, do
  - (a) Use $\hat{\mathbf{X}}_i^{(0)} = \mathbf{Y}_i$ as initial values, and iteratively find $\{\hat{\mathbf{X}}_i, \hat{\omega}_{ij}\} = \underset{\mathbf{X}_i, \omega_{ij}}{\arg\min} \sum_{j \in \mathbf{S}_i} \omega_{ij} \mathbb{D}_c^2(\mathbf{X}_i, \mathbf{Y}_j)$ by Eq. (9) and (10).
  - (b) Assign $\hat{\mathbf{X}}(i)$ as the center pixel value in $\hat{\mathbf{X}}_i$.

Output: Denoised image $\hat{\mathbf{X}}$.

3. EXPERIMENTS

To show the denoising performance of proposed UDNLFM, we compare UDNLFM with six NLM-based methods. They are traditional NLM, PNLM, LJS-NLM, NLEM, INLFM and NLFM. These denoising methods are applied to remove Gaussian noise from images. PSNR and SSIM [16] are used to quantitatively evaluate their denoising results.



(a) (b)

Fig. 1. Average values of PSNR and SSIM on all the test images: (a) average PSNR; (b) average SSIM.

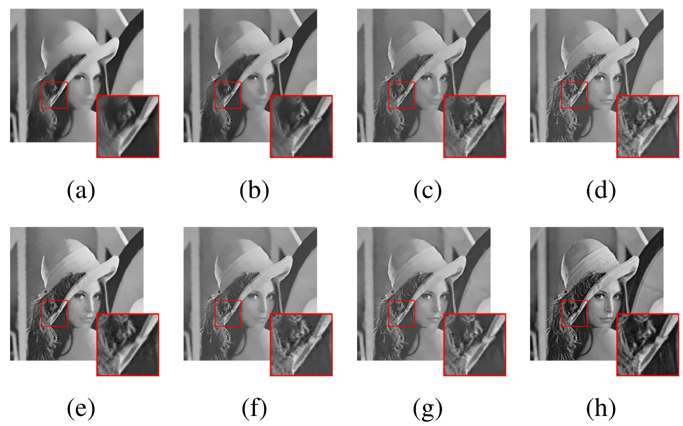


Fig. 2. Denoised results of the *lena* image with noise $\sigma = 20$: (a) NLM; (b) NLEM; (c) INLEM; (d) PNLM; (e) LJS-NLM; (f) NLFM; (g) UDNLFM; (h) Clean image.

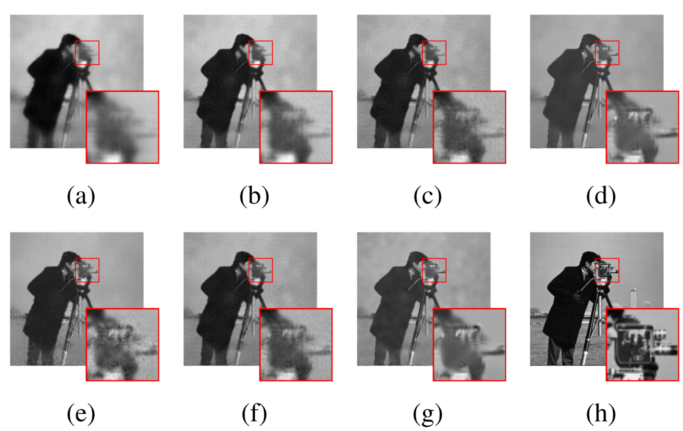


Fig. 3. Denoised results of the *cameraman* image with noise $\sigma = 60$: (a) NLM; (b) NLEM; (c) INLEM; (d) PNLM; (e) LJS-NLM; (f) NLFM; (g) UDNLFM; (h) Clean image.

Table 2. PSNR and SSIM results of UDNLFM and other methods at noise levels $\sigma = 10, 20, \cdots, 100$.

		PSNR(dB)										SSIM(%)									
Image	Method\\σ	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100
c.man	NLM	31.49	27.98	24.46	22.61	21.50	20.96	20.43	19.87	19.64	19.43	87.86	81.97	76.29	71.78	67.99	65.33	62.98	60.16	58.25	56.77
	NLEM	31.37	28.11	25.16	23.26	22.27	21.96	21.53	20.98	20.76	20.40	88.52	81.98	74.72	68.13	61.65	56.93	52.17	47.50	43.11	40.38
	INLEM	31.80	27.81	25.81	24.08	22.76	22.25	21.64	20.96	20.60	20.14	90.38	81.50	73.55	65.96	57.69	51.85	46.29	41.17	36.76	33.86
	PNLM	32.36	28.89	27.23	26.06	24.84	24.05	**22.97**	22.05	21.69	21.01	**92.49**	**85.85**	81.54	78.60	75.02	71.99	68.44	64.61	62.58	58.99
	LJS-NLM	**33.06**	29.36	26.94	25.25	23.77	22.97	22.09	21.20	20.78	20.20	92.30	85.83	79.49	74.92	69.20	64.70	60.09	54.72	50.63	46.97
	NLFM	32.15	28.14	26.16	24.75	23.38	22.72	21.99	21.23	20.88	20.40	91.60	82.72	75.58	69.24	61.51	55.74	50.13	44.87	40.28	37.27
	UDNLFM	32.98	**29.37**	**27.64**	**26.44**	**25.13**	**24.37**	22.41	**22.45**	**22.01**	**21.36**	92.21	85.70	**82.13**	**79.55**	**75.94**	**73.83**	**69.13**	**66.49**	**63.88**	**60.64**
lena	NLM	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
	NLEM	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
	INLEM	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
	PNLM	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	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
	NLFM	31.76	28.13	26.21	24.82	23.83	23.18	22.52	21.77	21.33	20.87	90.32	82.82	75.80	69.52	63.42	58.15	52.68	47.83	43.56	40.17
	UDNLFM	**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**
peppers	NLM	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
	NLEM	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
	INLEM	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
	PNLM	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	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
	NLFM	32.04	28.60	26.50	24.89	23.93	23.00	22.22	21.59	20.97	20.47	91.15	85.09	79.02	72.37	66.41	60.72	55.43	50.84	45.89	41.65
	UDNLFM	**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**
house	NLM	33.28	27.91	25.96	24.76	24.05	23.56	23.03	22.76	22.36	22.14	89.60	80.92	76.43	73.45	71.57	69.92	68.26	66.81	65.10	**63.81**
	NLEM	33.41	28.36	26.50	25.38	24.75	24.26	23.68	23.24	22.77	22.24	90.27	81.57	75.31	69.51	64.57	59.59	53.73	49.38	44.76	41.57
	INLEM	33.18	28.88	26.95	25.66	24.80	24.13	23.39	22.80	22.20	21.60	90.77	81.89	73.83	66.02	59.30	52.97	46.42	41.46	36.72	33.55
	PNLM	34.61	31.42	29.18	27.23	26.12	25.28	24.67	24.18	23.53	23.12	**92.81**	**87.84**	83.00	78.05	74.75	71.67	69.51	66.91	64.19	62.13
	LJS-NLM	34.70	30.49	28.10	26.34	25.33	24.57	23.84	23.31	22.70	22.33	92.69	85.52	79.50	73.90	69.81	65.24	61.06	56.86	52.81	49.84
	NLFM	33.71	29.90	27.67	26.12	25.17	24.47	23.73	23.15	22.57	21.99	91.54	84.19	76.64	69.07	62.74	56.71	50.38	45.49	40.68	37.45
	UDNLFM	**34.78**	**31.53**	**29.39**	**27.63**	**26.58**	**25.74**	**25.05**	**24.47**	**24.07**	**23.33**	92.63	87.57	**83.38**	**79.12**	**76.09**	**73.22**	**70.93**	**68.29**	**65.84**	63.71

We conduct the denoising experiments on forty grayscale images of size 256×256. The parameter settings of UDNLFM in the experiments are $k = 2$, $s = 7$, $h = 2.74\sigma$, $h_s = 255\sigma$, $\alpha = 0.5$ and $\beta = 3.0$. Since σ is needed in the calculation of unbiased distances, we use its rough estimation value. For other six methods, we set their parameters as recommended in the related references.

Table 2 displays the quantitative results of seven denoising methods. The noise level σ ranges from 10 to 100. From the PSNR results, UDNLFM outperforms other methods on almost all situations. In terms of SSIM, UDNLFM shows competitive performance over other methods, although it is a slightly poorer result than PNLM and LJS-NLM when $\sigma = 10$ or 20, because the fixed patch size can not adapt to all noise levels. Especially for low noise levels, the patch size is a little larger to process noise around edges. Fig. 1 presents the average value of PSNR and SSIM on the test images. According to average PSNR results (Fig. 1(a)), UDNLFM (red line) is about 0.5dB higher than PNLM (cyan line) and 1dB higher than NLFM (green line) on each noise level. That means UDNLFM achieves an obvious improvement in PSNR compared with other six methods. In Fig. 1(b), it can be observed that the average SSIM values of INLEM, NLEM, NLFM and LJS-NLM drop rapidly as σ increases. When $\sigma = 70$, they are almost reduced below 60%. However, UDNLFM reaches 60% when $\sigma = 100$. These results demonstrate that UDNLFM works better on heavy noise. In summary, UDNLFM has the best denoising results in quantitative analysis.

Fig. 2 shows the denoised results of the *lena* image ($\sigma = 20$). INLEM (Fig. 2(c)) and NLFM (Fig. 2(f)) still remain noise near the edge of hat, while NLM (Fig. 2(a)) and NLEM (Fig. 2(b)) over-smooth the details of the feather on the hat. Besides, PNLM (Fig. 2(d)) and LJS-NLM (Fig. 2(e)) leave obvious noise around the mouth and eyes. UDNLFM (Fig. 2(g)) shows superior ability in removing noise and preserving details.

Fig. 3 shows denoised results of the $cameraman$ image on a higher noise level ($\sigma = 60$). NLM (Fig. 3(a)) smooths the image excessively, and thus it is difficult to identify the structure of the camera in its denoising result. NLEM (Fig. 3(b)), INLEM (Fig. 3(c)) and NLFM (Fig. 3(f)) bring a lot of ringing artifacts to the denoised images. In the result of LJS-NLM (Fig. 3(e)), there are noise remaining around the camera and the tripod. PNLM (Fig. 3(d)) shows a good denoising performance, but it also brings a few artifacts around the camera. Compared to other methods, UDNLFM (Fig. 3(g)) achieves the best visual result.

In Python, on a 2.70 GHz Intel Core i7 processor, UDNLFM takes an average runtime of about 14 seconds to denoise images of size 256×256 using a search window size of 15×15.

4. DISCUSSION

In this study, we first introduced three distances, named pixel-pixel unbiased distance, patch-patch unbiased distance and combined unbiased distance. Compared with traditional Euclidean distance, they are more robust to measure the image patch similarity. Using these unbiased distances, we further proposed a novel Unbiased Distance based NLFM (UDNLFM). Similar with NLFM, it also regards the weight ω_{ij} as a fuzzy variable and updates in each denoising iteration. Experiments have shown that UDNLFM outperforms other competing NLM-based methods in image denoising.

References

[1] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[2] Marc Lebrun, Antoni Buades, and Jean-Michel Morel, "Implementation of the "non-local bayes" (nl-bayes) image denoising algorithm," *Image Processing On Line*, vol. 3, pp. 1–42, 2013.

[3] Lei Zhang, Weisheng Dong, David Zhang, and Guangming Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.

[4] Nicola Pierazzo, Marc Lebrun, Martin Rais, Jean-Michel Morel, and Gabriele Facciolo, "Non-local dual image denoising," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 813–817.

[5] Chaude Knaus and Matthias Zwicker, "Progressive image denoising," *IEEE Transactions on Image Processing*, vol. 23, pp. 3114–3125, 2014.

[6] Tao Dai, Chao-Bing Song, Ji-Ping Zhang, and Shu-Tao Xia, "Pmpa: A patch-based multiscale products algorithm for image denoising," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 4406–4410.

[7] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, vol. 2, pp. 60–65.

[8] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, "Non-local means denoising," *Image Processing On Line*, vol. 1, pp. 208–212, 2011.

[9] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, "A review of image denoising algorithms, with a new one," *SIAM Journal on Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[10] Kunal N. Chaudhury and Amit Singer, "Non-local euclidean medians," *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 745–748, 2012.

[11] Zhonggui Sun and Songcan Chen, "Analysis of non-local euclidean medians and its improvement," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 303–306, 2013.

[12] Yue Wu, Brain Tracey, Premkumar Natarajan, and Joseph P. Noonan, "Probabilistic non-local means," *IEEE Signal Processing Letters*, vol. 20, no. 8, pp. 763–766, 2013.

[13] Yue Wu, Brain Tracey, Premkumar Natarajan, and Joseph P. Noonan, "James-stein type center pixel weights for non-local means image denoising," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 411–414, 2013.

[14] Rushi Lan, Yicong Zhou, Yuan Yan Tang, and C. L. Philip Chen, "Image denoising using non-local fuzzy means," in *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, 2015, pp. 196–200.

[15] Enming Luo, Shengjun Pan, and Truong Nguyen, "Generalized non-local means for iterative denoising," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 260–264.

[16] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.