Probabilistic Non-Local Means

Yue Wu, Brian Tracey, Premkumar Natarajan and Joseph P. Noonan

***Abstract*—In this paper, we propose a so-called probabilistic non-local means (PNLM) method for image denoising. Our main contributions are: 1) we point out defects of the weight function used in the classic NLM; 2) we successfully derive all theoretical statistics of patch-wise differences for Gaussian noise; and 3) we employ this prior information and formulate the probabilistic weights truly reflecting the similarity between two noisy patches. Our simulation results indicate the PNLM outperforms the classic NLM and many NLM recent variants in terms of the peak signal noise ratio (PSNR) and the structural similarity (SSIM) index. Encouraging improvements are also found when we replace the NLM weights with the PNLM weights in tested NLM variants.**

***Index Terms*—Image Denoising, Non-Local Means, Probabilistic Modeling, Adaptive Algorithm**

I. Introduction

Non-local means (NLM) is a popular data-adaptive image denoising technique introduced by Buades *et al.* [1], [2]. This technique is proven to be effective in many image denoising tasks and analyzes images on a patch-by-patch basis. In the classic NLM, a 2D clean image $\mathbf{x} = \{x_l\}_{l\in\mathbb{I}}$ defined on the spatial domain $\mathbb{I}$ is assumed to be contaminated by identically and independently distributed (i.i.d.) zero-mean Gaussian noise with an unknown variance σ^2, *i.e.*

$$y_l = x_l + n_l, \text{ and } n_l \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

where y_l, x_l and n_l denote the noisy observation, the clean image pixel and the pixel noise, respectively. The NLM then estimates the clean pixel x_l by using a weighted sum of the noisy pixels within a prescribed search region $\mathbb{S}$, typically a square or a rectangular region:

$$\widehat{x_l} = \textstyle\sum_{k\in\mathbb{S}_l} w_{l,k} y_k / W_l \tag{2}$$

where each weight is computed by quantifying the similarity between two local patches (denoted as $\mathbb{P}$) around noisy pixels y_l and y_k as shown in Eq. (3),

$$w_{l,k} = \exp\big(-\textstyle\sum_{j\in\mathbb{P}}(y_{l+j} - y_{k+j})^2/h\big) \tag{3}$$

and the summation of all weights is denoted as

$$W_l = \textstyle\sum_{k\in\mathbb{S}_l} w_{l,k}. \tag{4}$$

Although the original NLM [1], [2] includes a weak Gaussian smoother, the weight (3) is a simplified version with similar performance that is also widely accepted [3].

Brian Tracey and Joseph P. Noonan are with the department of electrical and computer engineering, Tufts university, 161 College Ave, Medford, MA 02155. Premkumar Natarajan is with the department of speech, language and multimedia, Raytheon BBN technologies, 10 Moulton St., Cambridge, MA 02138. Yue Wu was also affiliated with Tufts university, but is now affiliated with Raytheon BBN technologies (e-mail: ywu@bbn.com).

Within the NLM framework, much progress has been made in recent years. Some authors have focused on fast NLM implementation [4], [5], while others have explored NLM parameter optimization [3], or have adjusted the NLM framework to achieve better performance [6], [7]. We notice that one shared interest of these three topics is the weight function of the NLM, which is the core of the NLM algorithm. Calculation of NLM weights is the most computationally expensive part of the algorithm and is related to many parameter optimization schemes. It has long been noticed that the NLM weight function is somewhat inadequate [8] because it tends to give non-zero weights to dissimilar patches. However, the reason behind this inadequacy has not yet been fully explored.

In this letter, we focus on the NLM weight function and propose a new probabilistic solution. The rest of the paper is organized as follows: Sec. II shows the defects of the NLM weights; Sec. III proposes our PNLM framework with new probabilistic weights; Sec. IV shows simulation results; and we conclude the letter in Sec. V.

II. Problems with the NLM Weight Function

The NLM weight (3) is considered as $w_{l,k} = \exp(-D_{l,k}/h')$, where h' is a translation of h in (3) and

$$D_{l,k} = \textstyle\sum_{j\in\mathbb{P}}(y_{l+j} - y_{k+j})^2/2\sigma^2 \tag{5}$$

is the patch difference between the patches around y_l and y_k. In this way, (5) can be interpreted as the standard quantitative χ^2 test to measure the similarity of the two samples [9]. The statistical interpretation of the exponential function used in (3) is not straightforward [9], although may be possible to relate it to Gaussian kernels used in probability density estimation. Nevertheless, this exponential function gives a larger weight to a pixel with a smaller patch difference (Fig. 1(a)). Intuitively, this idea is quite reasonable, as it means that the NLM relies more on pixels with smaller patch differences. However, we demonstrate below that this exponential function makes the NLM weights somewhat problematic.

From now on, we consider $D_{l,k}$ as a random variable (r.v.) and assume patches around x_l and x_k match perfectly, *i.e.*

$$\textstyle\sum_{j\in\mathbb{P}}(x_{l+j} - x_{k+j})^2 = 0. \tag{6}$$

If they are disjoint, then $D_{l,k} \sim \chi^2_{|\mathbb{P}|}$, where $|\cdot|$ denotes the cardinality of $\mathbb{P}$ (*i.e.* the number of pixels in $\mathbb{P}$). Fig. 1 shows this distribution on the right, with the corresponding NLM weight function on the left. It is clear that the NLM weight function gives two equally probable $D_{l,k}$s very different weights and that it fails to give the largest weight to the most probable case. For $D_{l,k}$s close to its expected value, the weight errors are not too large because the corresponding region in the exponential curve is almost linear with a moderate slope.

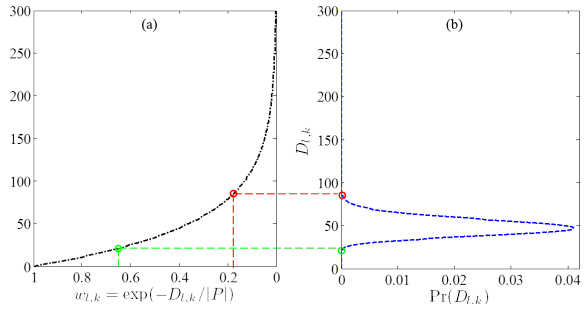

Fig. 1: NLM weight function and the distribution of 7×7 patch differences. (a) the NLM weight for $h=\sigma^2|\mathbb{P}|$; and (b) the distribution of disjoint patch differences. Red and green circles denotes two equal probable patch differences, while are biasedly weighted in NLM.

However, for $D_{l,k}$s far away from its expected value, weight errors are very large because the NLM function tends to give nonzero weights to these highly improbable cases, so weight errors grow quickly. In practice, correcting over-weighted weights has been shown to improve NLM performance. For example, the center pixel weight (CPW) in NLM is unitary and thus over-weights center pixels. [10], [11] report noticeable improvement just by tuning these over-weighted CPWs.

III. Probabilistic Non-Local Means

Instead of including the exponential function in weighting pixels, we propose the following probabilistic weight

$$w_{l,k}=f_{l,k}\big(\widehat{D}_{l,k}/\rho^2\big) \tag{7}$$

where $f_{l,k}(\cdot)$ is the theoretical probability density function (p.d.f.) of the r.v. $D_{l,k}$, $\widehat{D}_{l,k}$ is the estimated distance with estimated variance $\widehat{\sigma}^2$ in (5), and ρ is a tuning parameter. This weight function (7) can be interpreted as the probability of seeing a noisy patch difference when two clean patches match perfectly. Since it is clear we shall give a smaller weight in the more typical case when this perfectly matching condition fails, (7) then gives the largest similarity weight we shall consider.

A. Theoretical Distribution of Patch-wise Distance

Pretend we know the true noise variance σ^2 (later we will show this knowledge is unnecessary). Our goal is to derive the theoretical p.d.f. of the patch difference when the two clean patches around pixel x_l and x_k are perfectly matching (see (6)). To do so, we denote the pixel distance $d_{l,k}$ as

$$d_{l,k}=(y_l-y_k)^2/2\sigma^2 \tag{8}$$

and thus we have $D_{l,k}$ of the form that

$$D_{l,k}=\textstyle\sum_{j\in\mathbb{P}} d_{l+j,k+j}. \tag{9}$$

Because two patches are perfectly matching and noise is i.i.d., for all $j\in\mathbb{P}$ we have

$$d_{l+j,k+j}=\frac{(n_{l+j}-n_{k+j})^2}{2\sigma^2}\sim\chi^2_1.$$

If all $d_{l+j,k+j}$s are i.i.d., then we have $D_{l,k}\sim\chi^2_{|\mathbb{P}|}$, whose mean is $|\mathbb{P}|$ and variance is $2|\mathbb{P}|$. However, the i.i.d. assumption does not hold when the two patches overlap, as is the case for many pairs of patches. Fortunately, it is known that such a summed correlated χ^2 distribution can be well approximated [12] as,

$$D_{l,k}\sim\gamma_{l,k}\chi^2_{\eta_{l,k}} \tag{10}$$

where parameters $\gamma_{l,k}$ and $\eta_{l,k}$ can be determined by the first two cumulants of $D_{l,k}$ [12] as shown below.

$$\gamma_{l,k}=\mathrm{var}[D_{l,k}]/(2\mathrm{E}[D_{l,k}]) \tag{11}$$

$$\eta_{l,k}=\mathrm{E}[D_{l,k}]/\gamma_{l,k} \tag{12}$$

The cumulant $\mathrm{E}[D_{l,k}]$ is straightforward to find, and it is

$$\mathrm{E}[D_{l,k}]=\textstyle\sum_{j\in\mathbb{P}}\mathrm{E}[d_{l+j,k+j}]=|\mathbb{P}|. \tag{13}$$

With regards to $\mathrm{var}[D_{l,k}]$, the following identity always holds

$$\mathrm{var}[D_{l,k}]=\textstyle\sum_{i,j\in\mathbb{P}}\mathrm{cov}[d_{l+i,k+i},d_{l+j,k+j}] \tag{14}$$

where the covariance can be written as follows.

$$\mathrm{cov}[d_{l+i,k+i},d_{l+j,k+j}]=\mathrm{E}[d_{l+i,k+i}d_{l+j,k+j}]-\mu^2_{d_{l,k}} \tag{15}$$

This equation compares two pairs of r.v.s, $\mathbb{N}^i_{l,k}=\{n_{l+i},n_{k+i}\}$ and $\mathbb{N}^j_{l,k}=\{n_{l+j},n_{k+j}\}$, where l,k are distinctive patch center indices and i,j are location indices within the patch. Either 0, 1 or 2 of these values may be repeated. If n_l,n_k,n_u,n_v are distinctive noise observations from $\mathcal{N}(0,\sigma^2)$, it can be demonstrated that

$$\mathrm{E}[n_l^4/\sigma^4]=3 \tag{16}$$

$$\mathrm{E}[n_l^2n_k^2/\sigma^4]=1 \tag{17}$$

$$\mathrm{E}[n_ln_kn_un_v/\sigma^4]=\mathrm{E}[n_l^2n_kn_u/\sigma^4]=\mathrm{E}[n_l^3n_k/\sigma^4]=0 \tag{18}$$

This implies that, by expanding $\mathrm{E}[d_{l+i,k+i}d_{l+j,k+j}]$ terms, we can find its expectations under different conditions as follows.

$$\mathrm{E}[d_{l+i,k+i}d_{l+j,k+j}]=\begin{cases}3,\text{if }|\mathbb{N}^i_{l,k}\cap\mathbb{N}^j_{l,k}|=2\\1.5,\text{if }|\mathbb{N}^i_{l,k}\cap\mathbb{N}^j_{l,k}|=1\\1,\text{if }|\mathbb{N}^i_{l,k}\cap\mathbb{N}^j_{l,k}|=0\end{cases} \tag{19}$$

Since $\mathrm{var}[D_{l,k}]$ is the summation over the $|\mathbb{P}|\times|\mathbb{P}|$ covariance matrix, each term of which is dependent on the number of overlapping pixels $|\mathbb{N}^i_{l,k}\cap\mathbb{N}^j_{l,k}|$, $\mathrm{var}[D_{l,k}]$ can be simply found by computing the number of terms for each case in (19). Case $|\mathbb{N}^i_{l,k}\cap\mathbb{N}^j_{l,k}|=2$ happens only for $i=j$, *i.e.* covariance terms along the main diagonal, and thus there are $|\mathbb{P}|$ terms of this kind. Case $|\mathbb{N}^i_{l,k}\cap\mathbb{N}^j_{l,k}|=1$ happens for one pixel overlapping. Denote $\mathbb{O}_{l,k}$ as the set of overlapping pixels between two patches, then the number terms of this case is $2|\mathbb{O}_{l,k}|$, where multiplier 2 is from the symmetry of a covariance matrix and cardinality $|\mathbb{O}_{l,k}|$ is the number of overlapping pixels. Case $|\mathbb{N}^i_{l,k}\cap\mathbb{N}^j_{l,k}|=0$ happens for all disjoint pixel pairs. The number of terms of this type is $|\mathbb{P}|^2-|\mathbb{P}|-2|\mathbb{O}_{l,k}|$. These results, together with the fact that $\mu^2_{d_{l,k}}=1$, are used to find $\mathrm{var}[D_{l,k}]$ as

$$\mathrm{var}[D_{l,k}]=|\mathbb{P}|\cdot(3-1)+2|\mathbb{O}_{l,k}|\cdot(1.5-1)+0=2|\mathbb{P}|+|\mathbb{O}_{l,k}| \tag{20}$$

Since $\mathbb{O}_{l,k}$ is known once l and k are given, $\mathrm{var}[D_{l,k}]$ is then also known. As a result, $\gamma_{l,k}$ and $\eta_{l,k}$ in (11)-(12) can be fully determined, implying the p.d.f. of $D_{l,k}$ is

$$f_{l,k}(D)=\chi^2_{\eta_{l,k}}(D/\gamma_{l,k})=\frac{(D/\gamma_{l,k})^{\eta_{l,k}/2-1}\exp(-D/2\gamma_{l,k})}{2^{\eta_{l,k}/2}\Gamma(\eta_{l,k}/2)}. \tag{21}$$

The different spatial relationships of patch pairs imply different $|\mathbb{O}_{l,k}|$, thus causing different var$[D_{l,k}]$, $\gamma_{l,k}$, and finally p.d.f. $f_{l,k}$. This conclusion means that NLM weights calculated without considering spatial correlations are inadequate.

B. Parameters Discussions

Above we did not use $f_{l,k}(\widehat{D}_{l,k})$ as our weight function, but instead used $f_{l,k}(\widehat{D}_{l,k}/\rho^2)$. The parameter ρ^2 provides a way to adjust our probabilistic model when an estimated variance $\widehat{\sigma}^2$ is used instead of the true σ^2. When

$$\rho^2 = \sigma^2/\widehat{\sigma}^2 \tag{22}$$

reflects the ratio of the true noise variance to the estimated one, all previous derivations hold because

$$D_{l,k} = \frac{\widehat{\sigma}^2}{\sigma^2}\widehat{D}_{l,k} = \widehat{D}_{l,k}/\rho^2.$$

The raw probabilistic CPW $w_{l,l} = f_{l,k}(0) \approx 0$ under-weights a noisy center pixel. A more plausible CPW is

$$w_{l,l} = \chi^2_{|\mathbb{P}|}(|\mathbb{P}|). \tag{23}$$

which is the same as the weight of the most probable case. This CPW is used in the rest of the letter.

IV. Simulation Results

All of the following simulations are done under the MATLAB® r2012b environment. Our two goals are 1) to show that the derived p.d.f. $f_{l,k}$ in (21) closely approximates its true p.d.f.; and 2) to confirm the superiority of the proposed probabilistic weights and PNLM.

Fig. 2 shows the var$[D_{l,k}]$ map for a 7×7 search region $\mathbb{S}_l$ with 3×3 patches and the six typical theoretical p.d.f.s $f_{l,k}$, plotted with the corresponding sample distributions estimated from 100,000 realizations on a noise image $\mathbf{n}$, each pixel n_l follows an i.i.d. standard Gaussian $\mathcal{N}(0,1)$. It is noticeable that the var$[D_{l,k}]$ map is location-dependent and isotropic with one of the six theoretical values {18,19,20,21,22,24}. The more pixels overlap, the larger var$[D_{l,k}]$ is, implying a smaller peak on its p.d.f. It is clear that the predicted p.d.f.s are very close to those estimated from a large number of samples.

Since it is clear that the accuracy of the $f_{l,k}$ approximation degrades as correlation increases, the approximation accuracy of the most-overlapped cases can be used to characterize the worst-case accuracy. For each combination of search region $\mathbb{S}_l$ and patch size $\mathbb{P}$, there are four possible ks that attain the maximum correlation, all of which are one pixel away from the center pixel (see examples for k=18, 24, 26, and 32 in Fig. 2-(a)). In Table I, we report the averaged P-values of goodness of fit tests for the most correlated $f_{l,k}$s, where each P-value is the averaged from P-values of the four most correlated $f_{l,k}$s. Because all observed P-values are above 5%, we say the approximated theoretical p.d.f. (21) gives satisfactory predictions, so these p.d.f.s can reliably be used to quantify patch similarities.

In the following simulation, we compare the three pairs of NLM and PNLM algorithms, namely 1) the classic NLM and the proposed PNLM, 2) the classic NLM with the James-Stein Shrinkage (JSNLM) [11] and the proposed PNLM with the James-Stein Shrinkage (PSJNLM), and 3) the nonlocal median with the classic weights (NLEM) [6] and the nonlocal median with the probabilistic weights (PNLEM). The only difference between the two algorithms in each pair is the weight function. With regards to the parameter settings, we use patch size 7 and search region size 21 for all methods. For the temperature parameter h in NLMs, we use $h = |\mathbb{P}|\sigma^2$, which is nearly optimal and suggested in [3], and $\rho = 1$ is used in PNLMs. To quantify the quality of a denoised method, we compute the average PSNR [3] and SSIM [13] scores from 10 realizations for each method and each noise level. These results are reported in Table II.

From Table II, it is clear that 1) the proposed PNLM method outperforms the NLM method and those recent variants like NLEM and JSNLM; and 2) by replacing the NLM weight with the new proposed probabilistic one, both NLEM and JSNLM are improved in terms of higher PSNR/SSIM scores. Fig.3 gives sample denoising results and method noise images of the NLM and PNLM algorithms. These results show the effectiveness of the new proposed probabilistic weight and the superiority of the PNLM framework.

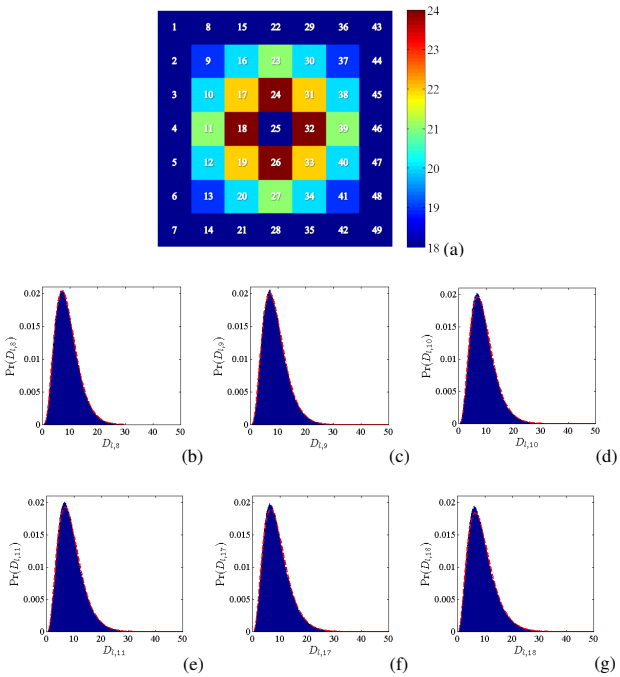

Fig. 2: Theoretical and estimated p.d.f. $f_{l,k}$s for 7×7 search region and 3×3 patches. (a) theoretical var$[D_{l,k}]$ map (in each cell, white indices indicate ks in $\mathbb{S}_l$, background colors represent the magnitude of var$[D_{l,k}]$). (b)-(g) theoretical (red dash lines) and estimated (blue bars) p.d.f. $f_{l,k}$s for $k=8$ (var$[D_{l,8}]$=18), $k=9$ (var$[D_{l,9}]=19$),$k=10$ (var$[D_{l,10}]=20$),$k=11$ (var$[D_{l,11}]=21$), $k=17$ (var$[D_{l,17}]=22$) and $k=18$ (var$[D_{l,18}]=24$), respectively.

TABLE I: Averaged P-values of goodness of fit tests for the observed sample distributions.

		Search Region Size				
		7	11	15	21	29
Patch Size	3	0.5853	0.6865	0.2252	0.1001	0.5612
	5	0.2675	0.3125	0.3374	0.5746	0.3501
	7	0.3967	0.4659	0.1545	0.2645	0.4282
	9	0.4741	0.8665	0.5233	0.3405	0.6582

TABLE II: Performance comparisons for NLM and PNLM methods

PSNR(dB)\ σ		10	20	30	40	50	60	70	80	90	100
cameraman	NLM	32.57	28.92	26.98	24.98	23.52	22.52	21.84	21.24	20.82	20.44
	PNLM	32.47	29.08	**27.44**	**26.26**	**25.19**	**24.13**	**23.26**	**22.44**	**21.84**	**21.31**
	NLEM	32.66	28.90	26.63	24.78	23.35	22.16	21.78	21.31	20.95	20.55
	PNLEM	**33.06**	**29.42**	27.36	25.72	24.90	23.87	23.11	22.28	21.73	21.13
	JSNLM	32.64	29.01	27.13	25.46	24.12	23.10	22.33	21.61	21.09	20.63
	PSJNLM	32.21	29.08	27.45	26.16	25.07	24.05	23.20	22.40	21.77	21.23
house	NLM	34.08	31.30	28.79	26.88	25.62	24.66	23.85	23.31	22.90	22.45
	PNLM	**34.92**	**32.40**	**30.48**	**28.70**	**27.25**	**26.14**	**24.98**	**24.17**	**23.57**	**22.98**
	NLEM	34.30	30.43	27.80	26.53	25.51	24.88	24.13	23.52	22.92	22.43
	PNLEM	34.56	31.97	30.24	28.64	27.07	26.11	24.90	24.14	23.38	22.94
	JSNLM	34.62	31.70	29.29	27.30	25.94	24.87	23.98	23.35	22.86	22.37
	PSJNLM	34.81	32.38	30.36	28.58	27.11	25.89	24.82	23.99	23.36	22.75
lenna	NLM	33.74	30.91	28.72	27.14	26.03	25.13	24.42	23.88	23.44	23.03
	PNLM	34.59	**32.07**	**30.17**	**28.58**	**27.32**	26.23	25.33	24.59	23.98	23.43
	NLEM	33.56	30.00	28.41	27.30	26.47	25.65	25.05	24.29	23.64	23.12
	PNLEM	33.78	31.21	29.64	28.32	27.26	**26.28**	**25.58**	**24.80**	**24.20**	**23.69**
	JSNLM	34.38	31.41	29.21	27.49	26.26	25.26	24.47	23.85	23.33	22.86
	PSJNLM	**34.72**	**32.07**	30.09	28.48	27.20	26.07	25.15	24.38	23.74	23.15
checker	NLM	39.04	33.80	30.95	28.94	27.37	25.94	24.45	23.25	21.84	20.87
	PNLM	**40.34**	**35.17**	**32.31**	**30.26**	**28.38**	**26.71**	**25.37**	**24.50**	**23.29**	**22.76**
	NLEM	39.68	34.13	30.90	28.94	27.07	25.62	24.59	23.54	22.66	22.09
	PNLEM	39.72	34.49	31.25	29.18	27.15	25.77	24.68	23.78	22.93	22.52
	JSNLM	39.03	33.79	30.93	28.93	27.35	25.91	24.42	23.22	21.81	20.84
	PSJNLM	34.64	30.86	31.23	29.73	27.95	26.39	25.08	24.25	23.10	22.60
SSIM(%)\ σ		10	20	30	40	50	60	70	80	90	100
cameraman	NLM	91.08	82.92	78.50	73.87	68.97	64.18	59.78	55.58	51.87	48.69
	PNLM	**91.64**	**84.65**	**80.23**	**76.61**	**73.26**	**69.72**	**66.18**	**62.72**	**59.45**	**56.60**
	NLEM	88.68	80.24	72.53	64.98	59.15	53.04	48.67	43.96	40.45	35.95
	PNLEM	91.16	83.22	78.13	73.31	68.89	63.11	58.90	54.51	50.54	46.18
	JSNLM	91.23	84.32	78.91	73.63	68.74	64.01	59.62	55.34	51.36	48.23
	PSJNLM	89.69	84.04	79.29	74.97	71.01	66.98	63.13	59.30	55.61	52.82
house	NLM	87.63	83.77	79.88	75.11	70.63	66.34	62.16	58.30	54.86	51.40
	PNLM	**89.38**	**85.00**	**81.72**	**78.18**	**74.70**	**71.05**	**67.43**	**63.99**	**60.81**	**58.05**
	NLEM	88.06	81.79	75.43	69.11	63.09	57.09	51.09	45.81	41.19	37.04
	PNLEM	89.17	84.11	79.92	75.17	70.40	65.40	59.92	54.59	49.58	46.80
	JSNLM	89.12	84.14	79.54	74.52	69.78	65.19	60.80	56.94	53.33	49.86
	PSJNLM	89.34	84.57	80.64	76.45	72.35	68.03	63.80	60.11	56.49	53.34
lenna	NLM	87.86	83.98	79.39	74.90	70.72	66.72	62.87	59.23	55.82	52.59
	PNLM	89.69	**85.00**	**81.18**	**77.56**	**74.16**	**70.78**	**67.57**	**64.51**	**61.44**	**58.81**
	NLEM	88.21	81.19	75.44	69.48	63.71	57.57	52.22	46.92	42.29	38.54
	PNLEM	89.56	84.02	79.03	74.05	69.42	64.64	60.09	55.85	51.87	48.25
	JSNLM	89.37	83.94	79.16	74.48	69.95	65.67	61.61	57.86	54.32	50.99
	PSJNLM	**89.75**	84.64	80.21	76.00	71.93	67.97	64.16	60.70	57.23	54.11
checker	NLM	99.01	97.38	95.35	93.23	90.75	87.94	84.22	80.76	76.31	72.00
	PNLM	**99.33**	**98.32**	**97.03**	**95.68**	**93.82**	**91.79**	**89.48**	**87.61**	**84.87**	**82.95**
	NLEM	99.10	97.67	95.46	92.65	89.20	85.12	81.90	78.73	74.26	71.71
	PNLEM	99.22	97.99	96.03	94.26	91.83	88.85	85.91	83.16	79.38	77.35
	JSNLM	99.00	97.34	95.24	93.11	90.53	87.59	83.79	80.23	75.65	71.18
	PSJNLM	94.25	92.72	95.09	93.74	91.66	89.53	86.86	84.83	82.02	80.14

V. Conclusion

In this letter, we pointed out the insufficiency of the NLM weights and showed a new promising PNLM framework, whose weights better reflect patch similarities. The proposed PNLM framework connects the denoising process and the noise type and thus is meaningful for denoising other types of noise. As long as a noise p.d.f. is known, we can estimate $f_{l,k}$ correspondingly. In this way, a universal denoising framework (see example in [14]) for multiple types of known noises and mixed noises may be developed. In addition, the proposed PNLM can also be extended to capture non i.i.d. noises, because one can easily to replace the p.d.f. of patch difference $f_{l,k}$ with more general forms. For example, for Gaussian noises with changing variance, $\int f_{l,k}(D|\sigma^2)\Pr(\sigma^2)\mathrm{d}\sigma^2$ can be used in place of $f_{l,k}(D)$. The proposed PNLM also provides a theoretical basis to quantify patch similarities, namely critical values $D_\alpha^{*\pm}$ such that $\Pr(D_\alpha^{*-}\leq D\leq D_\alpha^{*+}|f_{l,k})=\alpha$, which can be used as early terminations (see [4] for similar usage) and as thresholds to reject or accept a so-called similar patch, *e.g.* the empirical BM3D [15], [16] hard thresholds τ_{match} in Eq. (2) of [15]. A PNLM implementation is provided in MATLAB® central *file exchange* site (file ID: #41390).

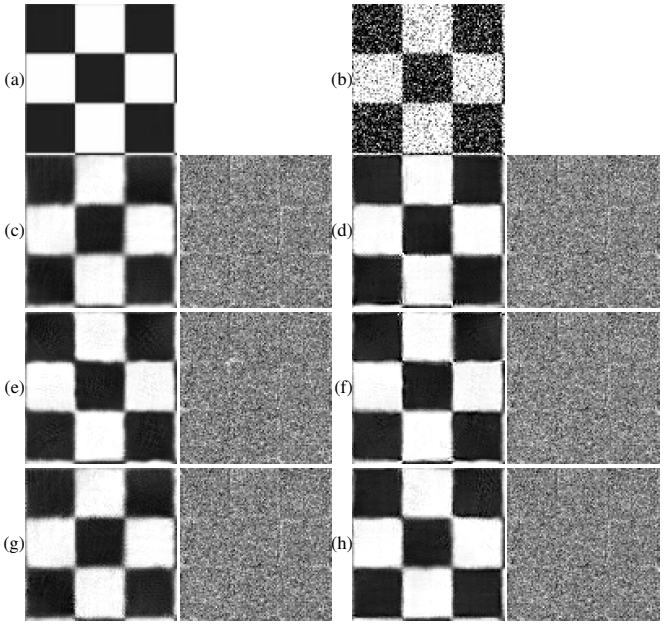

Fig. 3: NLM and PNLM denoising results for $\sigma=80$ (cropped and enlarged from results of image *checker*). (a) clean image; (b) noisy observation; (c) to (h): denoising results and method noise images of NLM, PNLM, NLEM, PNLEM, JSNLM, and PJSNLM, respectively.

References

[1] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. & Sim.*, vol. 4, no. 2, pp. 490–530, 2005.

[2] ——, "A non-local algorithm for image denoising," in *IEEE Comput. Soc. Conf. on Comput. Vision & Pattern Recognition*, vol. 2, june 2005, pp. 60–65.

[3] D. Van De Ville and M. Kocher, "Sure-based non-local means," *IEEE Signal Process. Lett.*, vol. 16, no. 11, pp. 973–976, 2009.

[4] R. Vignesh, B. T. Oh, and C.-C. Kuo, "Fast non-local means computation with probabilistic early termination," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 277–280, 2010.

[5] K. Chaudhury, "Acceleration of the shiftable o(1) algorithm for bilateral filtering and non-local means," *IEEE Trans. Image Process.*, no. 99, p. 1, 2012.

[6] K. Chaudhury and A. Singer, "Non-local euclidean medians," *IEEE Signal Process. Lett.*, vol. 19, no. 11, pp. 745–748, 2012.

[7] ——, "On the convergence of the irls algorithm in non-local patch regression," *accepted to IEEE Int. Conf. Acoust., Speech, and Signal Process.*, 2013.

[8] V. Duval, J. Aujol, and Y. Gousseau, "A bias-variance approach for the nonlocal means," *SIAM J. Imaging Sci.*, vol. 4, no. 2, pp. 760–788, 2011.

[9] N. Thacker, J. Manjon, and P. Bromiley, "Statistical interpretation of non-local means," *IET Comput. Vis.*, vol. 4, no. 3, pp. 162–172, 2010.

[10] J. Salmon, "On two parameters for denoising with non-local means," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 269–272, 2010.

[11] Y. Wu, B. Tracey, P. Natarajan, and J. Noonan, "James-stein type center pixel weights for non-local means image denoising," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 277–280, 2013.

[12] L.-L. Chuang and Y.-S. Shih, "Approximated distributions of the weighted sum of correlated chi-squared random variables," *J. Stat. Plan. Infer.*, vol. 142, no. 2, pp. 457–472, 2012.

[13] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[14] Z. Sun and S. Chen, "Modifying nl-means to a universal filter," *Opt. Commun.*, vol. 285, no. 24, pp. 4918–4926, 2012.

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3d filtering," in *Proceedings of SPIE*, vol. 6064, 2006, pp. 354–365.

[16] ——, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.