REFERENCES

[1] L. J. Karam, T. Ebrahimi, S. S. Hemami, T. N. Pappas, R. J. Safranek, Z. Wang, and A. B. Watson, "Introduction to the issue on visual media quality assessment," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 2, pp. 189–192, Apr. 2009.

[2] R. Ferzli and L. J. Karam, "A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB)," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 717–728, Apr. 2009.

[3] N. G. Sadaka, L. J. Karam, R. Ferzli, and G. P. Abousleman, "A noreference perceptual image sharpness metric based on saliency-weighted foveal pooling," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 369–372.

[4] R. Hassen, Z. Wang, and M. Salama, "No-reference image sharpness assessment based on local phase coherence measurement," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2010, pp. 2434–2437.

[5] J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field," *Vis. Res.*, vol. 21, no. 3, pp. 409–418, 1981.

[6] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "Perceptual blur and ringing metrics: Applications to JPEG2000," *Signal Process.: Image Commun.*, vol. 19, no. 2, pp. 163–172, Feb. 2004.

[7] N. D. Narvekar, "Objective no-reference visual blur assessment," M.S. thesis, Dept. Electr. Eng., Arizona State Univ., Tempe, 2009.

[8] Z. Liu, L. J. Karam, and A. B. Watson, "JPEG2000 encoding with perceptual distortion control," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1763–1778, Jul. 2006.

[9] D. Hood and M. Finkelstein, *Handbook of Perception and Human Performance*. New York: Wiley, 1986.

[10] H. R. Sheikh, A. C. Bovik, L. Cormack, and Z. Wang, "LIVE image quality assessment database," 2003 [Online]. Available: http://live.ece.utexas.edu/research/quality

[11] N. Ponomarenko, M. Carli, V. Lukin, K. Egiazarian, J. Astola, and F. Battisti, "Color image database for evaluation of image quality metrics," in *Proc. Int. Workshop Multimedia Signal Process.*, Oct. 2008, pp. 403–408.

[12] P. Le Callet and F. Autrusseau, "Subjective quality assessment IRCCyN/IVC database," 2005 [Online]. Available: http://www.irccyn.ec-nantes.fr/ivcdb/

[13] Z. M. P. Sazzad, Y. Kawayoke, and Y. Horita, "Image quality evaluation database," [Online]. Available: http://mict.eng.u-toyama.ac.jp/database_toyama/

[14] "Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment," VQEG, 2000.

[15] S. Tourancheau, F. Autrusseau, Z. M. Parvez, and Y. Horita, "Impact of subjective datasets on the performance of image quality metrics," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 365–368.

Nonlocal Means With Dimensionality Reduction and SURE-Based Parameter Selection

Dimitri Van De Ville, *Member, IEEE*, and Michel Kocher

***Abstract*—Nonlocal means (NLM) is an effective denoising method that applies adaptive averaging based on similarity between neighborhoods in the image. An attractive way to both improve and speed-up NLM is by first performing a linear projection of the neighborhood. One particular example is to use principal components analysis (PCA) to perform dimensionality reduction. Here, we derive Stein's unbiased risk estimate (SURE) for NLM with linear projection of the neighborhoods. The SURE can then be used to optimize the parameters by a search algorithm or we can consider a linear expansion of multiple NLMs, each with a fixed parameter set, for which the optimal weights can be found by solving a linear system of equations. The experimental results demonstrate the accuracy of the SURE and its successful application to tune the parameters for NLM.**

***Index Terms*—Linear transforms, nonlocal means (NLM), principal component analysis (PCA), Stein's unbiased risk estimate.**

I. INTRODUCTION

Learning from neighborhoods has become an important and powerful data-driven approach for various applications in image processing. Most notably, the nonlocal means (NLM) [1] algorithm applies adaptive averaging based on similar neighborhoods in a search region. Various methods have been proposed to accelerate the initial approach using preselection of the contributing neighborhoods based on average value and gradient [2], average and variance [3] or higher-order statistical moments [4], cluster tree arrangement [5], and [6], [7]. The computation of the distance measure between different neighborhoods itself can be optimized using the fast Fourier transform [8], a moving average filter [9], [10], early termination of the search [11], or by reducing redundant comparisons [12].

Variations of the NLM algorithm have also been proposed to improve the denoising performance; e.g., adaptive neighborhoods [13], iterative application [5], combination with kernel regression [14] and spectral analysis [15], and other similarity measures based on principal component analysis (PCA) [6], [16] or rotation invariance [17]. The smoothing parameter that determines the contributions of the patches has been locally optimized using Mallow's C_p statistic [18]. The most evolved version of the nonlocal principle is probably BM3D [19], which further processes the selected neighborhoods and gives high quality results.

The combination of NLM with dimensionality reduction methods such as PCA [6], [16] and SVD [7] has gained increased interest since the advantages are twofold. First, the computational complexity is highly reduced. Second, measuring the distance between neighborhoods in a lower-dimensional subspace improves robustness to noise;

Manuscript received August 11, 2010; revised December 06, 2010, February 10, 2011; accepted February 21, 2011. Date of publication March 07, 2011; date of current version August 19, 2011. This work has been funded in part by the Swiss National Science Foundation (PP00P2-123438, D. Van De Ville) and in part by the Centre for Biomedical Imaging (CIBM). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rafael Molina.

D. Van De Ville is with the Department of Radiology and Medical Informatics, University of Geneva, 1211 Geneva 14, Switzerland, and also with the Institute of Bioengineering, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015, Switzerland.

M. Kocher is with the Biomedical Imaging Group, EPFL, Lausanne 1015, Switzerland.

Digital Object Identifier 10.1109/TIP.2011.2121083

1057-7149/$26.00 © 2011 IEEE

e.g., results for NLM denoising for a 7×7 neighborhood are clearly improved by reducing the 49 dimensions to 5–10. Moreover, Tasdizen [20] proposed the combination of PCA and NLM with parallel analysis to select the dimensionality [21].

Stein's unbiased risk estimate (SURE) [22] is one elegant way to estimate the mean squared error (MSE) of an image degraded by additive Gaussian noise. Following this principle, one can select optimal parameters for regularization in inverse problems [23]–[25], in denoising strategies for wavelet thresholding [26]–[28], or using a numerical procedure for denoising approaches in general [29]. In recent work, we derived an analytical form of SURE [22] for the NLM algorithm [30]. This way, the MSE can be monitored from the noisy image only, which is a very useful property to optimally tune the NLM algorithm. This concept can also be used to locally adapt the NLM parameters [12].

Here we further extend the analytical form of the SURE for NLM with linear projection of the neighborhoods, including projection on a dimensionality-reduced subspace as specified by PCA. Since the PCA-NLM algorithm depends nonlinearly on the different parameters (neighborhood size, width of smoothing kernel, search region, PCA dimensionality), we propose to optimize a linear expansion of several NLMs with different parameter settings; an approach that is inspired by the SURE-based linear expansion of thresholds (LET) proposed for wavelet denoising [31]. In our case, the optimal linear combination can be retrieved using the SURE of the individual PCA-NLM contributions.

In Section II, we briefly review the NLM algorithm and the SURE principle, together with the extension of SURE for NLM with linear projection. We also show how a linear expansion of multiple NLMs reduces to solving a linear system of equations. Next, in Section III, we present and discuss the experimental results to demonstrate the feasibility of using SURE for NLM parameter selection.

II. Methods

A. Nonlocal Means Algorithm

We consider the observation model

$$\mathbf{y} = \mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^N$ stands for the vector representation of the noise-free image containing N pixels, $\mathbf{n}$ is the zero-mean white Gaussian noise of variance $\sigma^2\mathbf{I}$, and $\mathbf{y}$ is the observed noisy data. We denote the grayscale value of the individual pixel at position $\mathbf{l} \in \mathcal{I}$ as $y_{\mathbf{l}}$, where we implicitly assume that vector indexing is mapped to a scalar index (e.g., using lexicographic ordering); this notation better reflects the spatial dependencies of the image. The pixel-based NLM algorithm [1] is a spatially adaptive filter that maps the measured data $\mathbf{y}$ into $\hat{\mathbf{x}}$ as follows:

$$\hat{x}_{\mathbf{l}} = \frac{\sum_{\mathbf{k}\in\mathcal{S}_{\mathbf{l}}} w_{\mathbf{k},\mathbf{l}} y_{\mathbf{k}}}{\sum_{\mathbf{k}\in\mathcal{S}_{\mathbf{l}}} w_{\mathbf{k},\mathbf{l}}} \tag{2}$$

where $\mathcal{S}_{\mathbf{l}}$ is the search region around $\mathbf{l}$ and $w_{\mathbf{k},\mathbf{l}}$ are the weights that compare the neighborhoods around pixels $\mathbf{k}$ and $\mathbf{l}$, respectively. The weights are defined as

$$w_{\mathbf{k},\mathbf{l}} = \exp\left(-\frac{\sum_{\mathbf{b}\in\mathcal{B}}(y_{\mathbf{k}+\mathbf{b}} - y_{\mathbf{l}+\mathbf{b}})^2}{B\,2\lambda^2}\right) \tag{3}$$

where $\mathcal{B}$ defines the neighborhood and B is its total size; e.g., $\mathcal{B} = [-3,3] \times [-3,3]$ and $B = 49$ for a 7×7 neighborhood.

B. Mean Squared Error and Stein's Unbiased Risk Estimate

The mean squared error (MSE) of the denoised image with respect to its noise-free version is

$$\mathrm{MSE}(\hat{\mathbf{x}}) = \frac{1}{N}||\mathbf{x} - \hat{\mathbf{x}}||^2 = \frac{1}{N}\sum_{\mathbf{l}\in\mathcal{I}}(x_{\mathbf{l}} - \hat{x}_{\mathbf{l}})^2 \tag{4}$$

where $||\cdot||^2$ is the Euclidean norm. The peak signal-to-noise ratio (PSNR) is then defined as

$$\mathrm{PSNR}(\mathrm{MSE}(\hat{\mathbf{x}})) = -10\log_{10}\frac{\mathrm{MSE}(\hat{\mathbf{x}})}{\mathrm{peak}^2} \tag{5}$$

where the denominator indicates the peak intensity value of the image. SURE provides a means for unbiased estimation of the true MSE. It is specified by the following analytical expression [22]

$$\mathrm{SURE}(\hat{\mathbf{x}}) = \frac{1}{N}||\mathbf{y} - \hat{\mathbf{x}}||^2 - \sigma^2 + 2\sigma^2\frac{\mathrm{div}_{\mathbf{y}}\{\hat{\mathbf{x}}\}}{N} \tag{6}$$

where $\mathrm{div}_{\mathbf{y}}\{\hat{\mathbf{x}}\}$ is the divergence of the NLM algorithm with respect to the measurements

$$\mathrm{div}_{\mathbf{y}}\{\hat{\mathbf{x}}\} = \sum_{\mathbf{l}\in\mathcal{I}}\frac{\partial\hat{x}_{\mathbf{l}}}{\partial y_{\mathbf{l}}} \tag{7}$$

which needs to be well defined in the weak sense. The derivation of SURE relies on the additive white Gaussian noise hypothesis and assumes the knowledge of the noise variance σ^2. In practice, σ^2 can be easily estimated from the measured data (e.g., using the median of absolute deviation). The SURE-based PSNR, which we will name SURE-PSNR from now on, can then be computed as $\mathrm{PSNR}(\mathrm{SURE}(\hat{\mathbf{x}}))$.

In previous work, we derived the analytical form of SURE for NLM [30]

$$\begin{aligned}\mathrm{SURE}(\hat{\mathbf{x}}) &= \frac{1}{N}||\mathbf{y} - \hat{\mathbf{x}}||^2 - \sigma^2 \\ &\quad + \frac{2\sigma^2}{NB\lambda^2}\left(\sum_{\mathbf{l}\in\mathcal{I}}\widehat{x_{\mathbf{l}}^2} - ||\hat{\mathbf{x}}||^2\right) + \frac{2\sigma^2}{N}\sum_{\mathbf{l}\in\mathcal{I}}\frac{1}{W_{\mathbf{l}}} \\ &\quad + \frac{2\sigma^2}{NB\lambda^2}\sum_{\mathbf{l}\in\mathcal{I}}\frac{1}{W_{\mathbf{l}}}\sum_{\mathbf{b}\in\mathcal{B}} w_{\mathbf{l}-\mathbf{b},\mathbf{l}}(y_{\mathbf{l}} - y_{\mathbf{l}+\mathbf{b}})(\hat{x}_{\mathbf{l}} - y_{\mathbf{l}-\mathbf{b}})\end{aligned} \tag{8}$$

where $W_{\mathbf{l}} = \sum_{\mathbf{k}\in\mathcal{S}_{\mathbf{l}}} w_{\mathbf{k},\mathbf{l}}$ and where $\widehat{\mathbf{x}^2}$ is the NLM algorithm applied to the squared pixel values. The computation of the divergence term can be readily incorporated within the core of the NLM algorithm. Specifically, implementing (8) requires an additional memory array to store $\widehat{x_{\mathbf{l}}^2}$ (next to $W_{\mathbf{l}}$), and its computational complexity takes only $O(B \cdot N)$ operations, compared to $O(B \cdot N \cdot S)$ of the NLM algorithm itself, where S is the number of pixels in the search region.

C. Nonlocal Means for Transformed Neighborhoods

Instead of using directly the pixel values of the neighborhoods as positions in the high-dimensional space, an appealing alternative is to first transform the neighborhoods in another domain with some favorable properties. For example, the computational burden of the NLM algorithm can be alleviated by projecting the neighborhood into a subspace of lower dimensionality as determined by PCA [6], [16], [20]. Specifically, the projection matrix $\mathbf{A}$ that diagonalizes the demeaned covariance matrix of all patches in the image is computed. Then, each neighborhood centered around $\mathbf{k}$ can be projected onto the vector $\mathbf{p}_{\mathbf{k}} = [p_{\mathbf{k},n}]_{n=1,\ldots,B'}$ that is in a subspace with B' dimensions with $B' \leq B$

$$p_{\mathbf{k},n} = \sum_{\mathbf{b}\in\mathcal{B}} a_{n,\mathbf{b}} y_{\mathbf{k}+\mathbf{b}}, \quad n = 1,\ldots,B'. \tag{9}$$

The only adaptation to the NLM algorithm is to redefine the weights as

$$w_{\mathbf{k},\mathbf{l}} = \exp\left(-\frac{\sum_{n=1}^{B'}(p_{\mathbf{k},n} - p_{\mathbf{l},n})^2}{B'\,2\lambda^2}\right). \tag{10}$$

The use of dimensionality reduction ($B' < B$) can significantly speed-up the algorithm. Here we extend the derivation of the SURE for the case of PCA-based NLM. The first step consists of deriving the divergence term.

Fig. 1. Results for the "peppers256" test image corrupted with additive white noise of $\sigma = 50$. (a) Original image. (b) Noisy image. (c) Results obtained with BM3D [19]. Without dimensionality reduction (no PCA): (d) 1 NLM, exhaustive optimization; (e) 12 NLMs with fixed parameters, optimal linear expansion using SURE; (f) 12 NLMs with Monte-Carlo generated parameters, optimal linear expansion using SURE. With dimensionality reduction (6 PCs): (g) 1 NLM, exhaustive optimization; (h) 12 NLMs with fixed parameters, optimal linear expansion using SURE; (i) 12 NLMs with Monte-Carlo generated parameters, including PCA dimensionality B', optimal linear expansion using SURE.

Proposition 1 (Divergence of NLM With Linear Transform): The individual terms of the divergence $\mathrm{div}_{\mathbf{y}}\{\hat{\mathbf{x}}\}$ of the NLM algorithm after transforming the neighborhoods according to (9) are given by

$$\frac{\partial \hat{x}_{\mathbf{l}}}{\partial y_{\mathbf{l}}} = \frac{1}{W_{\mathbf{l}}} \left(\sum_{n=1}^{B'} a_{n,\mathbf{0}} \sum_{\mathbf{k} \in \mathcal{S}_{\mathbf{l}}} \frac{w_{\mathbf{k},\mathbf{l}}}{B'\lambda^2} y_{\mathbf{k}} p_{\mathbf{k},n} + 1 - \hat{x}_{\mathbf{l}} \sum_{n=1}^{B'} a_{n,\mathbf{0}} \sum_{\mathbf{k} \in \mathcal{S}_{\mathbf{l}}} \frac{w_{\mathbf{k},\mathbf{l}}}{B'\lambda^2} p_{\mathbf{k},n} + \frac{1}{B'\lambda^2} \sum_{\mathbf{b} \in \mathcal{B}} w_{\mathbf{l}-\mathbf{b},\mathbf{l}} \times \sum_{n=1}^{B'} (p_{\mathbf{l}-\mathbf{b},n} - p_{\mathbf{l},n}) a_{n,\mathbf{b}} (\hat{x}_{\mathbf{l}} - y_{\mathbf{l}-\mathbf{b}}) \right). \quad (11)$$

Given our vector-indexing, it is important to note that the element $a_{n,\mathbf{0}}$ correspond to the weight of the projection matrix $\mathbf{A}$ for the center position in the neighborhood contributing to the projection on the nth component.

Proposition 2 (SURE for NLM With Linear Transform): The SURE for the NLM algorithm can be expressed as

$$\mathrm{SURE}(\hat{\mathbf{x}}) = \frac{1}{N}\|\mathbf{y} - \hat{\mathbf{x}}\|^2 - \sigma^2 + \frac{2\sigma^2}{NB'\lambda^2} \sum_{\mathbf{l} \in \mathcal{I}} \frac{1}{W_{\mathbf{l}}} \left(\sum_{n=1}^{B'} a_{n,\mathbf{0}} \sum_{\mathbf{k} \in \mathcal{S}} w_{\mathbf{k},\mathbf{l}} y_{\mathbf{k}} p_{\mathbf{k},n} + B'\lambda^2 - \hat{x}_{\mathbf{l}} \sum_{n=1}^{B'} a_{n,\mathbf{0}} \sum_{\mathbf{k} \in \mathcal{S}} w_{\mathbf{k},\mathbf{l}} p_{\mathbf{k},n} + \sum_{\mathbf{b} \in \mathcal{B}} w_{\mathbf{l}-\mathbf{b},\mathbf{l}} \times \sum_{n=1}^{B'} (p_{\mathbf{l}-\mathbf{b},n} - p_{\mathbf{l},n}) a_{n,\mathbf{b}} (\hat{x}_{\mathbf{l}} - y_{\mathbf{l}-\mathbf{b}}) \right). \quad (12)$$

Fig. 2. Results for the "lena512" test image corrupted with additive white noise of $\sigma = 20$. (a) Noisy image. (b) Results obtained with BM3D [19], PSNR 33.05 dB. (c) Best result obtained with the proposed method using dimensionality reduction (6 PCs) and 12 NLMs with Monte-Carlo generated parameters, including PCA dimensionality B', optimal linear expansion using SURE, PSNR 32.53 dB.

Note that this expression is valid for any linear transformation of the neighborhood, with or without dimensionality reduction. The computational complexity for obtaining SURE has increased with respect to (8); i.e., it is now $O(B' \cdot N \cdot S)$, which is of the same order as the NLM algorithm. However, the operations of (12) can be incorporated in the core loop of the NLM algorithm.

D. Selection of Best NLM

Using the proposed SURE for NLM, we are able to compare the performance of NLMs with different parameter sets (B, B', S, λ), in order to improve the denoising capabilities.

E. Linear Expansion of Multiple NLMs

Another possibility is inspired by the approach from [31]: we consider the linear combination of the outputs of several NLMs with and different fixed parameter sets, and we optimize these linear weights by SURE to hopefully exceed the performance of each NLM taken individually. Specifically, we consider the linear expansion approach as

$$\hat{\mathbf{x}} = \sum_{m=1}^{M} c_m \hat{\mathbf{x}}^{(m)}. \tag{13}$$

In our case, $\hat{\mathbf{x}}^{(m)}$ is the mth NLM with parameter set $(B_m, B'_m, S_m, \lambda_m)$ and c_m is the weight in the linear expansion. The optimal weights are obtained by minimizing the SURE of the linear combination. From (6)

$$\mathrm{SURE}(\hat{\mathbf{x}}) = \frac{1}{N}(\hat{\mathbf{x}} - \mathbf{y})^T(\hat{\mathbf{x}} - \mathbf{y}) - \sigma^2 + \frac{2\sigma^2}{N}\mathrm{div}_{\mathbf{y}}\{\hat{\mathbf{x}}\} \tag{14}$$

the partial derivatives towards the weights c_m are then given by

$$\frac{\partial \mathrm{SURE}(\hat{\mathbf{x}})}{\partial c_m} = \sum_{l=1}^{M} c_l \hat{\mathbf{x}}^{(m)T}\hat{\mathbf{x}}^{(l)} - \mathbf{y}^T\hat{\mathbf{x}}^{(m)} + \sigma^2 \mathrm{div}_{\mathbf{y}}\left\{\hat{\mathbf{x}}^{(m)}\right\} \tag{15}$$

which leads to the following system of equations:

$$\sum_{l=1}^{M} c_l \hat{\mathbf{x}}^{(m)T}\hat{\mathbf{x}}^{(l)} = \mathbf{y}^T\hat{\mathbf{x}}^{(m)} - \sigma^2 \mathrm{div}_{\mathbf{y}}\left\{\hat{\mathbf{x}}^{(m)}\right\}, m = 1, \ldots, M \tag{16}$$

where the derivation of the SURE provides us with the divergence terms. We can then find the linear weights that optimize the SURE of the linear expansion efficiently.

III. Results and Discussion

We describe how SURE for NLM can be successfully deployed for automatic parameter selection following various optimization strategies. In the various experiments, the parameter space that we will sample from is as follows:

- neighborhood $\mathcal{B} = 3 \times 3, 5 \times 5, 7 \times 7$, so $B = 9, 25, 49$;
- dimensionality of projection $B' = 6, 7, \ldots, B$;
- search region $\mathcal{S} = 5 \times 5, 7 \times 7, \ldots, 21 \times 21$, so $S = 25, 49, 441$;
- smoothing parameter $\lambda/\sigma = 0.5, 0.6, \ldots, 1.2$.

All results discussed in detail below are summarized in the Table II. We also show some visual examples for the "peppers" test image, Fig. 1(a), corrupted with additive Gaussian noise of $\sigma = 50$, Fig. 1(b), and the "lena" test image corrupted with noise of $\sigma = 20$, Fig. 2(a).

TABLE I
Heuristic Choice of Parameters for Each of the NLM When Performing Linear Expansion of Multiple Ones

	neighborhood $\mathcal{B}$	search region $\mathcal{S}$	smoothing λ/σ
1	3×3	21×21	0.7
2	5×5	21×21	0.7
3	7×7	21×21	0.7
4	3×3	21×21	0.4
5	5×5	21×21	0.4
6	7×7	21×21	0.4
7	3×3	5×5	1.0
8	5×5	5×5	1.0
9	7×7	5×5	1.0
10	3×3	11×11	0.85
11	5×5	11×11	0.85
12	7×7	11×11	0.85

A. Exhaustive Optimization

As a starting point, we perform an exhaustive optimization for a single NLM to determine the best parameters (B^*, S^*, λ^*) for $B' = B$ (which corresponds to no PCA dimensionality reduction) and $B' = 6$, respectively. The global optimum within this parameter space is found by choosing the settings corresponding to the best SURE-PSNR, which coincides with the optimal setting for the ground-truth PSNR for all test images and noise levels. Moreover, SURE-PSNR was always close within 0.10 dB to the true one—see the results in the rows "1 NLM" in Table II. The best parameter setting of the NLM varied with the test image and with the noise level, which indicates the importance of a data-adaptive strategy such as obtained using SURE. It is surprising

TABLE II
PERFORMANCE OF THE VARIOUS APPROACHES AS MEASURED BY PSNR AND SURE-PSNR (BETWEEN PARENTHESES)

	cameraman (256×256)					
	$\sigma = 20$			$\sigma = 50$		
method	no PCA	6 PCs	variable PCs	no PCA	6 PCs	variable PCs
1 NLM	29.65 (29.51) [1]	29.82 (29.66) [1]		24.59 (24.45) [1]	25.09 (25.06) [1]	
1 NLM + self [2]	29.73 (29.62) [1]	29.84 (29.72) [1]	29.82 (29.68) [4]	25.07 (24.89) [1]	25.10 (25.07) [1]	25.40 (25.25) [4]
3 NLM + self [2,3]	29.69 (29.60)	29.81 (29.67)		24.88 (24.71)	25.47 (25.35)	
6 NLM + self [2,3]	29.76 (29.69)	29.91 (29.80)		24.99 (24.84)	25.53 (25.39)	
12 NLM + self [2,3]	29.97 (29.87)	30.04 (29.92)		25.09 (25.07)	25.60 (25.49)	
3 NLM + self [2,4]	29.81 (29.63)	30.02 (29.88)	29.99 (29.82)	25.19 (24.94)	25.46 (25.30)	25.53 (25.36)
6 NLM + self [2,4]	29.90 (29.78)	30.08 (29.94)	30.03 (29.91)	25.43 (25.21)	25.52 (25.39)	25.64 (25.52)
12 NLM + self [2,4]	29.66 (30.17)	30.09 (29.97)	30.09 (29.95)	25.20 (25.56)	25.59 (25.52)	25.79 (25.91)
BM3D		30.48 (N/A)			25.84 (N/A)	
	house (256×256)					
	$\sigma = 20$			$\sigma = 50$		
method	no PCA	6 PCs	variable PCs	no PCA	6 PCs	variable PCs
1 NLM	32.02 (32.00) [1]	32.81 (32.84) [1]		26.37 (26.42) [1]	28.41 (28.45) [1]	
1 NLM + self [2]	32.11 (32.13) [1]	32.81 (32.84) [1]	32.78 (32.82) [4]	26.97 (27.04) [1]	28.41 (28.47) [1]	28.06 (28.37) [4]
3 NLM + self [2,3]	32.20 (32.22)	32.36 (32.40)		26.69 (26.80)	27.86 (28.11)	
6 NLM + self [2,3]	32.20 (32.23)	32.73 (32.81)		26.78 (26.88)	28.27 (28.55)	
12 NLM + self [2,3]	32.33 (32.44)	32.60 (33.25)		27.05 (27.46)	27.52 (30.70)	
3 NLM + self [2,4]	32.28 (32.30)	32.62 (33.00)	32.79 (32.77)	27.17 (27.33)	28.01 (28.59)	27.84 (28.01)
6 NLM + self [2,4]	32.48 (32.44)	32.78 (33.17)	32.90 (32.94)	27.49 (27.71)	28.00 (29.32)	27.85 (28.15)
12 NLM + self [2,4]	32.23 (32.84)	32.67 (33.50)	32.98 (33.01)	27.72 (28.01)	27.37 (30.45)	28.20 (28.50)
BM3D		33.77 (N/A)			29.37 (N/A)	
	peppers (256×256)					
	$\sigma = 20$			$\sigma = 50$		
method	no PCA	6 PCs	variable PCs	no PCA	6 PCs	variable PCs
1 NLM	30.37 (30.37) [1]	30.71 (30.72) [1]		25.41 (25.35) [1]	26.51 (26.53) [1]	
1 NLM + self [2]	30.49 (30.49) [1]	30.71 (30.72) [1]	30.74 (30.73) [4]	25.73 (25.67) [1]	26.51 (26.54) [1]	26.47 (26.54) [4]
3 NLM + self [2,3]	30.09 (30.14)	30.42 (30.43)		25.20 (25.17)	26.17 (26.32)	
6 NLM + self [2,3]	30.09 (30.15)	30.64 (30.71)		25.27 (25.25)	26.38 (26.53)	
12 NLM + self [2,3]	30.85 (30.87)	31.01 (31.09)		26.05 (26.10)	26.53 (26.99)	
3 NLM + self [2,4]	30.76 (30.76)	30.83 (30.87)	30.97 (30.95)	26.03 (26.02)	26.51 (26.56)	26.48 (26.49)
6 NLM + self [2,4]	30.80 (30.79)	31.05 (31.09)	31.02 (31.01)	26.34 (26.36)	26.43 (26.79)	26.59 (26.65)
12 NLM + self [2,4]	30.73 (31.37)	31.05 (31.13)	31.06 (31.25)	26.45 (26.47)	26.52 (27.00)	26.70 (27.00)
BM3D		31.29 (N/A)			26.41 (N/A)	
	Lena (512×512)					
	$\sigma = 20$			$\sigma = 50$		
method	no PCA	6 PCs	variable PCs	no PCA	6 PCs	variable PCs
1 NLM	31.54 (31.59) [1]	32.14 (32.17) [1]		26.94 (26.89) [1]	28.33 (28.30) [1]	
1 NLM + self [2]	31.65 (31.67) [1]	32.15 (32.18) [1]	32.03 (32.10) [4]	27.30 (27.27) [1]	28.33 (28.30) [1]	28.19 (28.20) [4]
3 NLM + self [2,3]	31.39 (31.43)	31.94 (31.97)		27.02 (27.00)	28.06 (28.05)	
6 NLM + self [2,3]	31.41 (31.46)	32.14 (32.15)		27.08 (27.05)	28.28 (28.23)	
12 NLM + self [2,3]	31.98 (31.98)	32.34 (32.34)		27.57 (27.59)	28.39 (28.37)	
3 NLM + self [2,4]	31.94 (31.95)	32.26 (32.29)	32.28 (32.33)	27.39 (27.38)	28.54 (28.50)	28.46 (28.42)
6 NLM + self [2,4]	31.91 (32.12)	32.34 (32.38)	32.39 (32.42)	27.88 (27.89)	28.55 (28.52)	28.39 (28.52)
12 NLM + self [2,4]	31.77 (32.29)	32.40 (32.43)	32.53 (32.56)	27.14 (28.49)	28.60 (28.57)	28.59 (28.61)
BM3D		33.05 (N/A)			28.86 (N/A)	

[1]: exhaustive optimisation of the NLM parameters
[2]: optimisation of the linear expansion using SURE
[3]: heuristic choice of NLM parameters according to Table I
[4]: Monte-Carlo generation of the NLM parameters (best out of 10 realizations)

that dimensionality reduction of the patches onto 6 principal components (PCs) improved the performance with 0.5–1.1 dB. Visually, this difference is also striking as can be observed by comparing Fig. 1(d) and (g). Moreover, next to the performance gain, the computational complexity of PCA-NLM with 6 PCs is reduced with almost one order of magnitude (factor of 6/49).

One way to improve the NLM method is to change the weight of the central pixel, which is overestimated in the classical NLM formulation; i.e., $w_{\mathbf{i},\mathbf{i}}$ is always at the maximum of one, which is independent of the noise level since the same two noise realizations are compared. This possibility has been mentioned by various authors and solved in different ways; e.g., the NLM weights can be estimated using the SURE principle as in [32]. Here, we propose to add the original (noisy) image to the set of images of the linear expansion. This image's weight will be determined by SURE and turns out to be negative in practice in order to lower the importance of the central pixel[1]. This way we easily obtain the optimal weights of the two contributions that result in the best performance. Despite the face that providing the noisy image improves the results for the NLM without dimensionality reduction, especially for high noise levels, the PCA-NLM method does not improve by doing so. This can be explained by the fact that the projection onto the most important components of the patches automatically also removes the sole influence of the central weight because none of the PC vectors will be localized at the central pixel of the patch. These results are listed in the rows "1 NLM + self" in Table II.

[1]Providing the noisy image itself to the linear expansion corresponds to adding the output of the identity operator, for which holds $\mathrm{div}_{\mathbf{y}}(\hat{\mathbf{x}}) = N$.

B. Heuristic Optimization

Given the improvement by adding a proportion of the original noisy image with the weight determined by the SURE, it is tempting to add more NLMs to the linear expansion. However, finding the optimal parameters of all NLMs jointly becomes unfeasible. Therefore, in the next series of experiments, we verify how the performance can be further improved by linearly combining the outputs of multiple NLMs, each one with predefined parameter set. In particular, when more NLMs are added, the parameter set is chosen according to Table I; i.e., we used 3, 6, and 12 NLMs, respectively, each time together with the original noisy image. The results are listed in Table II in the rows with the label[(3)]. For the case without dimensionality reduction, increasing

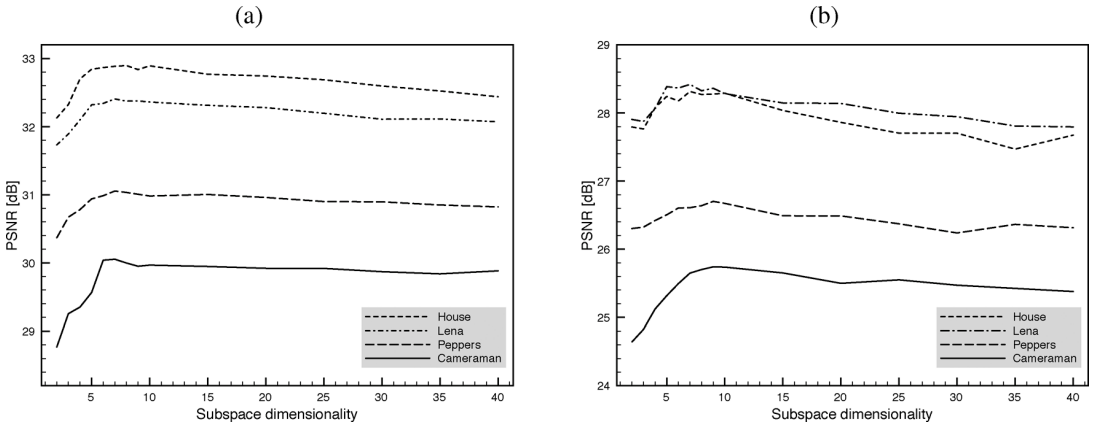

Fig. 3. Average PSNR (with respect to ground truth) over 10 realizations of NLM linear expansions with 12 NLM and Monte-Carlo generated parameters (B, S, λ). The SURE-PSNR (not indicated) is very close to the true PSNR (within 0.10 dB). (a) Noise level of $\sigma = 20$. (b) Noise level of $\sigma = 50$.

the number of NLMs always improved results. For the case with dimensionality reduction, the improvement from 6 to 12 NLMs becomes less significant. In some cases (e.g., "house" with dimensionality reduction), the SURE-PSNR improved but the true PSNR decreased. We believe that this is due to an overfitting of the linear expansion, especially for a simple image such as "house" where the neighborhoods have a relatively low dimensionality. In this case, the difference between the true PSNR and SURE-PSNR increases, as well as the dynamic range of the weights of the linear expansion (typically below 0.50). An example of "peppers" using 12 NLMs is shown in Fig. 1(e) and (h), without and with dimensionality reduction, respectively.

C. Monte-Carlo Optimization

From the optimal weights of the NLM linear expansions, we could not identify a clear trend that would be indicative for the "right" NLM parameters to use; i.e., there is a large variability for different images and noise levels. Therefore, we consider another experiment where all NLM parameters (B, B', S, λ) are randomly generated according to a uniform distribution within the range of neighborhood size, dimensionality of the neighborhood after PCA, search region, and smoothing parameter as defined in the beginning of Section III.

1) Selection of Best NLM: We now take one step back and select the best performing NLM, according to its SURE, for 120 realizations. This reverts parameter optimization by random sampling of the parameter space (B, B', S, λ). The results are indicated in the rows "$1\ \mathrm{NLM} + \mathrm{self}$" with label$^{(4)}$ in Table II. We observe that in almost all cases the performance from the exhaustive search (with fixed $B' = 6$) is not reached. This suggests that random sampling of the parameter space remains suboptimal, despite the high number (120) of realizations.

2) Linear Expansion of Multiple NLMs: We have seen before, when combining multiple NLMs with heuristic parameter sets with SURE-based linear expansion, that the *diversity* of the various contributions is more important than their individual quality. Therefore, we use now random parameters for 3, 6, and 12 NLMs. These are indicated in Table II with the label$^{(4)}$. Each time, the best performance of 10 Monte-Carlo realizations as indicated by the SURE-PSNR is reported. Interestingly, this simple method outperforms both the best single NLM and the NLM linear expansion with fixed parameters: combining only 3 NLMs often reached or improved the results over 12 NLMs with fixed parameters. The results for "peppers" are shown in Fig. 1(f) and (i). Despite the improved PSNR, visual observation of the images reveals a grainy appearance which is probably due to contributions of NLMs with small neighborhoods. As a comparison, the result obtained by the state-of-the-art algorithm BM3D [19], which uses basis functions that are better adapted to edges, is shown in (c). For natural images such as "lena," this difference is less obvious; e.g., see Figs. 2(b) and (c).

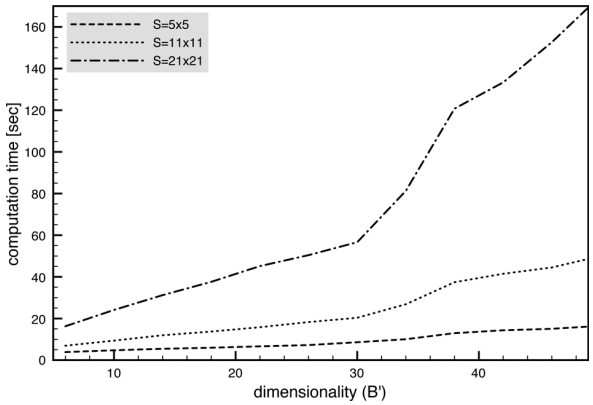

Fig. 4. Computation time of a single NLM algorithm for a 256×256 image as a function of the neighborhood dimensionality (B') and search region.

D. On the Dimensionality Reduction

We have observed that using PCA to reduce the dimensionality of the neighborhood is beneficial for both quality and computational complexity. The optimal number of dimensions is still dependent on the image content and the noise level, which explains why the linear expansion of NLMs with Monte-Carlo generated parameters is advantageous. In Fig. 3, we plot the average PSNR as a function of the number of dimensions for a linear expansion of 12 NLMs with (B, S, λ) Monte-Carlo generated.

E. On the Computational Complexity

Finally, we also briefly mention the computational complexity of the proposed method, which was implemented in Matlab (R2010b) using C for the core calculations (Intel Core 2 Duo, 2.66 GHz; 4 GB RAM). The dimensionality of the neighborhood (eventually after projection) and the search region are the two main parameters that influence the computation time. Therefore, in Fig. 4, we plot the computation time of a single NLM for a 256×256 image as a function of B' and S. Compared to the main NLM algorithm, the computational load of the divergence term for the SURE calculation and the optimal weights of the linear expansion (when combining multiple NLMs) are negligible.

$$\frac{\partial w_{\mathbf{k},\mathbf{l}}}{\partial y_{\mathbf{l}}} = \begin{cases} \frac{w_{\mathbf{k},\mathbf{l}}}{B'\lambda^2}\left(\sum_{n=1}^{B'}(p_{\mathbf{k},n}-p_{\mathbf{l},n})a_{n,\mathbf{0}}\right), & \mathbf{l}-\mathbf{k}\notin\mathcal{B} \\ \frac{w_{\mathbf{k},\mathbf{l}}}{B'\lambda^2}\left(\sum_{n=1}^{B'}(p_{\mathbf{k},n}-p_{\mathbf{l},n})a_{n,\mathbf{0}}\right) & \\ \quad -\frac{w_{\mathbf{k},\mathbf{l}}}{B'\lambda^2}\left(\sum_{n=1}^{B'}(p_{\mathbf{k},n}-p_{\mathbf{l},n})a_{n,\mathbf{l}-\mathbf{k}}\right), & \mathbf{l}-\mathbf{k}\in\mathcal{B} \end{cases} \tag{18}$$

Note that our best results taking the best realization of 10 Monte-Carlo parameter sets for 12 NLMs require 120 NLM evaluations, which typically adds up to 15–20 minutes. However, this type of algorithm can make use of a parallel implementation in an almost trivial way.

IV. Conclusion

We derived the SURE for the NLM algorithm with linear projection of the neighborhoods. The key feature of this derivation is the explicit analytical form of the divergence term of NLM, which is surprising for a nonlinear algorithm. The SURE can be easily computed on-the-fly as part of the original NLM algorithm.

The parameter setting of NLM is dependent on image content and noise level. Therefore, the SURE is a useful measure to estimate and tune these parameters. Next to exhaustive optimisation, we considered a linear expansion of multiple NLMs. We obtained the best performance for a linear combination of 12 NLMs using Monte-Carlo generated parameter sets. These results are close to the state-of-the-art denoising schemes while relying on the relatively simple algorithm of NLM and the SURE-based optimisation of linear weights.

Future work could further investigate the optimal structure of the NLM parameter settings. Promising avenues also include the use of different linear projections of the neighborhoods (e.g., to improve invariance to some features) and further development of a spatially adaptive version of NLM, including speeding up the algorithm [12]. Finally, the reprojection method from [33] could be incorporated to improve visual quality once the optimal parameter set determined.

Appendix A
Derivation of the Divergence Term

To obtain the divergence term, we introduce $W_{\mathbf{l}} = \sum_{\mathbf{k}\in\mathcal{S}_{\mathbf{l}}} w_{\mathbf{k},\mathbf{l}}$ and we derive $\hat{x}_{\mathbf{l}}$ with respect to $y_{\mathbf{l}}$, which results into

$$\begin{aligned} \frac{\partial \hat{x}_{\mathbf{l}}}{\partial y_{\mathbf{l}}} &= \frac{W_{\mathbf{l}}\frac{\partial}{\partial y_{\mathbf{l}}}(W_{\mathbf{l}}\hat{x}_{\mathbf{l}}) - (W_{\mathbf{l}}\hat{x}_{\mathbf{l}})\frac{\partial W_{\mathbf{l}}}{\partial y_{\mathbf{l}}}}{W_{\mathbf{l}}^2} \\ &= \frac{1}{W_{\mathbf{l}}}\left(\sum_{\mathbf{k}\in\mathcal{S}_{\mathbf{l}}}\frac{\partial w_{\mathbf{k},\mathbf{l}}}{\partial y_{\mathbf{l}}}y_{\mathbf{k}} + \underbrace{w_{\mathbf{l},\mathbf{l}}}_{=1} - \hat{x}_{\mathbf{l}}\sum_{\mathbf{k}\in\mathcal{S}_{\mathbf{l}}}\frac{\partial w_{\mathbf{k},\mathbf{l}}}{\partial y_{\mathbf{l}}}\right). \end{aligned} \tag{17}$$

Further on, deriving the weights gives (18), shown at the top of the page. By using the previous relations, we derive the constituting term of the divergence as (11). The divergence is finally given by combining the previous relation with (7).

Acknowledgment

The authors would like to thank Dr. G. Peyré for making available to the community his Matlab/C implementation of the NLM algorithm. The modified source code is available on request.

References

[1] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *SIAM Interdisciplinary J.: Multiscale Model. Simulat.*, vol. 4, no. 2, pp. 290–530, 2005.

[2] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 839–842, Dec. 2005.

[3] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images," *IEEE Trans. Med. Imag.*, vol. 27, no. 4, pp. 425–441, Apr. 2008.

[4] A. Dauwe, B. Goossens, H. Luong, and W. Philips, "A fast non-local image denoising algorithm," in *Proc. SPIE Electron. Imag.*, 2008, vol. 6812, pp. 681210–681210-8.

[5] T. Brox, O. Kleinschmidt, and D. Cremers, "Efficient nonlocal means for denoising of textural patterns," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1083–1092, Jul. 2008.

[6] N. Azzabou, N. Paragios, and F. Guichard, "Image denoising based on adapted dictionary computation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2007, vol. 3, pp. 109–112.

[7] J. Orchard, M. Ebrahimi, and A. Wong, "Efficient nonlocal-means denoising using the SVD," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2008, pp. 1732–1735.

[8] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, "Fast non-local algorithm for image denoising," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2006, pp. 1429–1432.

[9] B. Goossens, H. Luong, A. Pizurica, and W. Philips, "An improved non-local denoising algorithm," in *Proc. Int. Workshop Local Non-Local Approximat. Image Process. (LNLA)*, 2008, pp. 143–156.

[10] J. Darbon, A. Cunha, T. F. Chan, S. Osher, and G. J. Jensen, "Fast nonlocal filtering applied to electron cryomicroscopy," in *Proc. 5th IEEE Int. Symp. Biomed. Imag.: Nano Macro (ISBI2008)*, Paris, France, May 14–17, 2008, pp. 1331–1334.

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

[12] V. Duval, J. F. Aujol, and Y. Gousseau, On the Parameter Choice for the Non-Local Means HAL, Tech. Rep. HAL-00468856, 2010.

[13] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

[14] P. Chatterjee and P. Milanfar, "A generalization of non-local means via kernel regression," in *Proc. SPIE Electron. Imag.*, 2008, vol. 6814, pp. 68140P–68140P-9.

[15] G. Peyré, "Image processing with non-local spectral bases," *SIAM Multiscale Model. Simulat.*, vol. 7, no. 2, pp. 703–730, 2008.

[16] T. Tasdizen, "Principal components for non-local means image denoising," in *Proc. 15th IEEE Int. Conf. Image Process. (ICIP)*, San Diego, CA, 2008, pp. 1728–1731.

[17] S. Zimmer, S. Didas, and J. Weickert, "A rotationally invariant block matching strategy improving image denoising with non-local means," in *Proc. Int. Workshop Local Non-Local Approximat. Image Process.*, 2008, pp. 135–142.

[18] V. Doré and M. Cheriet, "Robust NL-means filter with optimal pixel-wise smoothing parameter for statistical image denoising," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1703–1716, May 2009.

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

[20] T. Tasdizen, "Principal neighborhood dictionaries for nonlocal means image denoising," *IEEE Trans. Image Process.*, vol. 18, no. 12, pp. 2649–2660, Dec. 2009.

[21] J. L. Horn, "A rationale and test for the number of factors in factor analysis," *Psychometrica*, vol. 30, no. 2, pp. 179–185, 1965.

[22] C. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 9, pp. 1135–1151, 1981.

[23] K. C. Li, "From Stein^{\prime}s unbiased risk estimates to the method of generalized cross validation," *Ann. Statist.*, vol. 13, no. 4, pp. 1352–1377, 1985.

[24] V. Solo, "A SURE-fired way to choose smoothing parameters in ill-conditioned inverse problems," in *ICIP*, 1996, vol. 3, pp. 89–92.

[25] C. Vonesch, S. Ramani, and M. Unser, "Recursive risk estimation for non-linear image deconvolution with a wavelet-domain sparsity constraint," in *Proc. 2008 IEEE Int. Conf. Image Process. (ICIP2008)*, San Diego, CA, Oct. 12–15, 2008, pp. 665–668.

[26] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995.
[27] A. Benazza-Benyahia and J.-C. Pesquet, "Building robust wavelet estimators for multicomponent images using Stein^{\prime}s principle," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1814–1830, Nov. 2005.
[28] F. Luisier, T. Blu, and M. Unser, "A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 593–606, Mar. 2007.
[29] S. Ramani, T. Blu, and M. Unser, "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008.
[30] D. Van De Ville and M. Kocher, "SURE-based non-local means," *IEEE Signal Process. Lett.*, vol. 16, no. 11, pp. 973–976, Nov. 2009.
[31] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778–2786, Nov. 2007.
[32] J. Salmon, "On two parameters for denoising with non-local means," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 269–272, Mar. 2010.
[33] J. Salmon and Y. Strozecki, "From patches to pixels in non-local methods: Weighted-average reprojection," in *Proc. 2010 IEEE 17th Int. Conf. Image Process.*, 2010, pp. 1929–1932.

Fast Bilateral Filter With Arbitrary Range and Domain Kernels

Bahadir K. Gunturk, *Senior Member, IEEE*

***Abstract*—In this paper, we present a fast implementation of the bilateral filter with arbitrary range and domain kernels. It is based on the histogram-based fast bilateral filter approximation that uses uniform box as the domain kernel. Instead of using a single box kernel, multiple box kernels are used and optimally combined to approximate an arbitrary domain kernel. The method achieves better approximation of the bilateral filter compared to the single box kernel version with little increase in computational complexity. We also derive the optimal kernel size when a single box kernel is used.**

***Index Terms*—Image enhancement, nonlinear filtering.**

I. Introduction

The bilateral filter is a nonlinear weighted averaging filter, where the weights depend on both the spatial distance and the intensity distance with respect to the center pixel. The main feature of the bilateral filter is its ability to preserve edges while doing spatial smoothing. The term "bilateral filter" was first used by Tomasi and Manduchi in [1]; the same filter was earlier called the SUSAN (Smallest Univalue Segment Assimilating Nucleus) filter by Smith and Brady in [2]. The variants of the bilateral filter have been published even earlier as the sigma filter [3] and the neighborhood filter [4].

Manuscript received April 14, 2010; revised November 02, 2010; accepted February 18, 2011. Date of publication March 14, 2011; date of current version August 19, 2011. This work was supported in part by the National Science Foundation under Grant 0528785 and the National Institutes of Health under Grant 1R21AG032231-01. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Luminita Aura Vese.

The author is with the Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: bahadir@ece.lsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2126585

At a pixel location $\mathbf{x} = (x_1, x_2)$, the output of the bilateral filter is calculated as follows:

$$\hat{I}(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} K_d(\|\mathbf{y} - \mathbf{x}\|) K_r(|I(\mathbf{y}) - I(\mathbf{x})|) I(\mathbf{y}) \quad (1)$$

where $K_d(\cdot)$ is the spatial domain kernel, $K_r(\cdot)$ is the intensity range kernel, $\mathcal{N}(\mathbf{x})$ is the set of pixels within a spatial neighborhood of $\mathbf{x}$, and $C(\mathbf{x})$ is the normalization term

$$C(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} K_d(\|\mathbf{y} - \mathbf{x}\|) K_r(|I(\mathbf{y}) - I(\mathbf{x})|). \quad (2)$$

The kernels $K_d(\cdot)$ and $K_r(\cdot)$ determine how the spatial and intensity differences are treated. The contribution (weight) of a pixel $I(\mathbf{y})$ is determined by the product of $K_d(\cdot)$ and $K_r(\cdot)$. The bilateral filter in [1] uses the Gaussian kernel, $G_\sigma(z) = \exp(-z^2/2\sigma^2)$, for both the domain and range kernels:

$$K_d(\|\mathbf{y} - \mathbf{x}\|) = G_{\sigma_d}(\|\mathbf{y} - \mathbf{x}\|) \quad (3)$$

and

$$K_r(|I(\mathbf{y}) - I(\mathbf{x})|) = G_{\sigma_r}(|I(\mathbf{y}) - I(\mathbf{x})|). \quad (4)$$

On the other hand, the sigma filter [3] and the neighborhood filter [4] use different kernels. The sigma filter [3] first calculates the local standard deviation around $I(\mathbf{x})$; the standard deviation is then used to determine a threshold value for pixel intensities, and pixels that are within the threshold of the center pixel $I(\mathbf{x})$ are averaged (with equal weights) to calculate the filter output at that pixel. In case of the neighborhood filter [4], the range kernel is a Gaussian as in (3), and the spatial kernel is a uniform box kernel. Among different kernel options, the Gaussian kernel is the most popular choice for both the range and spatial kernels, as it gives an intuitive and simple control of the behavior of the filter with two parameters, σ_d and σ_r.

The bilateral filter has found a wide range of applications in image processing and computer vision. The immediate application of the bilateral filter is image denoising as it can do spatial averaging without blurring edges. [5] presents a multiresolution extension of the bilateral filter for image denoising and an empirical study on optimal parameter selection. It is shown that the optimal value of σ_d is relatively insensitive to noise power, while the optimal σ_r value is linearly proportional to the noise standard deviation. Other applications of bilateral filter include tone mapping in high-dynamic range imaging [6], contrast enhancement [7], [8], fusion of flash and no-flash images [9], [10], fusion of visible spectrum and infrared spectrum images [11], compression artifact reduction [12], 3-D mesh denoising [13], [14], depth map estimation [15], video stylization [16], video enhancement [17], texture and illumination separation [18], orientation smoothing [19], and optical flow estimation [20].

This paper presents a fast approximation of the bilateral filter with arbitrary range and domain kernels. It is based on a method presented by Porikli in [21]. The method in [21] (which uses a box domain kernel) is extended by optimally combining multiple box kernels to approximate an arbitrary domain kernel. As there is no restriction on the range kernel either, any range and domain kernels can be used with this fast bilateral filter implementation. Section II reviews the fast bilateral filter techniques in the literature. The proposed method is explained in Section III. In Section IV, the question of optimal kernel size in case of a single box kernel is addressed. Section V provides some experimental results, and Section VI concludes the paper.

1057-7149/$26.00 © 2011 IEEE