# Bounded self-weights estimation method for non-local means image denoising using minimax estimators

Nguyen, Minh Phuong; Chun, Se Young

**DOI**
10.1109/TIP.2017.2658941

**Publication date**
2017

**Document Version**
Accepted author manuscript

**Published in**
IEEE Transactions on Image Processing

# Bounded Self-Weights Estimation Method for Non-Local Means Image Denoising Using Minimax Estimators

Minh Phuong Nguyen and Se Young Chun, *Member, IEEE*



***Abstract*— A non-local means (NLM) filter is a weighted average of a large number of non-local pixels with various image intensity values. The NLM filters have been shown to have powerful denoising performance, excellent detail preservation by averaging many noisy pixels, and using appropriate values for the weights, respectively. The NLM weights between two different pixels are determined based on the similarities between two patches that surround these pixels and a smoothing parameter. Another important factor that influences the denoising performance is the self-weight values for the same pixel. The recently introduced local James-Stein type center pixel weight estimation method (LJS) outperforms other existing methods when determining the contribution of the center pixels in the NLM filter. However, the LJS method may result in excessively large self-weight estimates since no upper bound is assumed, and the method uses a relatively large local area for estimating the self-weights, which may lead to a strong bias. In this paper, we investigated these issues in the LJS method, and then propose a novel local self-weight estimation methods using direct bounds (LMM-DB) and reparametrization (LMM-RP) based on the Baranchik's minimax estimator. Both the LMM-DB and LMM-RP methods were evaluated using a wide range of natural images and a clinical MRI image together with the various levels of additive Gaussian noise. Our proposed parameter selection methods yielded an improved bias-variance trade-off, a higher peak signal-to-noise (PSNR) ratio, and fewer visual artifacts when compared with the results of the classical NLM and LJS methods. Our proposed methods also provide a heuristic way to select a suitable global smoothing parameters that can yield PSNR values that are close to the optimal values.**

***Index Terms*— James-Stein estimator, minimax estimator, non-local means, center pixel weight, bounded self-weight, image denoising.**

## I. Introduction

IMAGE denoising is a fundamental task in image processing, low-level computer vision, and medical imaging algorithms. The goal of denoising is to suppress image noise when restoring desired details using prior information about the images. For example, based on prior information regarding "smooth images", a simple filter, such as a Gaussian filter, can be designed as a weighted average of the image intensities of the pixels in the local neighborhood with non-adaptive weights. However, this type of filter blurs the edges and details of images because these features are not captured in the assumed prior information. Many edge-preserving denoising methods have been proposed, including bilaterial filters [1], [2], anisotropic diffusion [3], non-local means (NLM) filters [4], [5], collaborative filters (BM3D) [6], and total variation filters [7]. Many filters, including bilaterial filters, anisotropic diffusion, and NLM filters (but, not BM3D, see [8]), can be represented as the weighted averages of adaptive weights or adaptive smoothing [9]. It should be noted that it is important to select appropriate weights in these types of filters in order to obtain improved denoised image quality [8].

Classical NLM filters use the similarities between two local patches in a noisy image to determine the weights in non-local adaptive smoothing [4]. The NLM weights are obtained by first calculating the Euclidean distance between the two local patches, which is denoted $d$, and then by evaluating $\exp(-d^2/h^2)$, where $h$ is a smoothing parameter. This method allows higher weights to be assigned to pixels with similar patches so that edges and details can be preserved through non-local weighted averaging.

There are four different factors that determine the output image quality of a NLM filter in terms of weights. 1) The first factor is the similarity measure $d$. The Euclidean distance is a usual choice [4], but other similarity measures have also been proposed, such as hypothesis testing with adaptive neighborhoods [10], principal component analysis (or the subspace based method) [11], [12], blockwise aggregation [13], rotation-invariant measures [14]–[16], shape-adaptive patches [17], and patch-based similarities with adaptive neighborhoods [18]. In multimodal medical imaging, inaccurate weights for noisy molecular images were enhanced by using additional high quality anatomical images [19], [20]. 2) The second factor is the strategy for determining the smoothing parameter $h$. Optimization strategies have been developed based on Stein's unbiased risk estimation (SURE) method for NLM with Gaussian noise [21], [22], NLM with Poisson noise [23], and blockwise NLM with Gaussian noise [24]. 3) The third factor is in selecting the function to use to determine the weights,

Manuscript received March 9, 2016; revised October 26, 2016 and December 31, 2016; accepted January 21, 2017. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning under Grant NRF-2014R1A1A1007928. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Oleg V. Michailovich. *(Corresponding author: Se Young Chun)*

Minh Phuong Nguyen was with the School of Electrical and Computer Engineering (ECE), Ulsan National Institute of Science and Technology (UNIST), Ulsan, 44919, South Korea. Now he is with Section of Radiation, Detection and Medical Imaging, Delft University of Technology, Mekelweg 15, 2629 JB Delft, The Netherlands (e-mail: m.p.nguyen@tudelft.nl).

Se Young Chun is with the School of ECE, UNIST, Ulsan, 44919, South Korea (e-mail: sychun@unist.ac.kr).

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.2017.2658941

such as $\exp(-x^2)$. Other functions have also been proposed to calculate the weights, such as compact support functions [25], [26] and statistical distance functions [27], [28]. 4) The last factor, which is the focus of this article, is in how best to determine the self-weights for the same pixel in the input and output images.

The NLM weights for two different pixels are essentially determined by the distance between the two noisy local patches around these pixels. However, the weights for the same pixel, or the self-weights, are not affected by the noise in the patches and the distance is always 0. For an extremely noisy image, the self-weights will be relatively too large when compared to the other weights, which will cause the filter output to be almost the same as the input noisy image. Therefore, the use of appropriate self-weight values can significantly affect the quality of the denoised image. Many researchers have investigated strategies for determining the self-weights, which are also known as center pixel weights, in order to alleviate the so-called "rare patch effect." For the classical NLM filter proposed by Buades *et al.*, the self-weights were set to be either one or the maximum weight in a neighborhood [4]. This strategy guaranteed that at least one or two of the largest weights would be the same. Doré and Cheriet also used the maximum weight in a neighborhood as the self-weight, but only if that maximum weight was large [29]. Brox and Cremers proposed a method to have at least $n$ number of the weights to be the same [30], and Zimmer *et al.* considered the self-weight to be a free parameter during the estimation process [31]. Salmon developed a SURE-based method for determining the self-weights that accounted for the noise [32].

Recently, Wu *et al.* proposed a method to determine the self-weights using a James-Stein (JS) type estimator [33]. The idea of that work was to use a JS estimator to determine the reparametrized self-weight in a local neighborhood (called the local JS estimator (LJS)). The LJS method yielded the best peak signal-to-noise ratio (PSNR) results when compared to other existing self-weight selection methods [4], [32]. However, the method had some limitations. First, the LJS could yield self-weights that were theoretically much larger than 1 because no upper bound for the self-weights was assumed, and this may lead to severe rare patch artifacts. The JS estimator does not guarantee its optimality for bounded shrinkage parameters. Second, the original LJS method was tested with a relatively large local neighborhood when determining a self-weight because it was assumed that the self-weights were the same in the local neighborhood. However, the problem is that the selection of a local neighborhood size that is too large may introduce a strong bias into the resulting denoised images.

In this article, we investigate the original LJS method in terms of the local neighborhood size for self-weight estimation and the potential for excessive self-weight estimation when no upper bound is applied on the self-weight. We then propose novel self-weight estimation methods for NLM that account for bounded self-weights using Baranchik's minimax estimator [34], called local minimax self-weight estimation with direct bound (LMM-DB) and with reparametrization (LMM-RP). We evaluated our proposed methods using performance criteria including PSNR, the bias-variance tradeoff curve and visual quality assessment with a wide range of natural images and a real patient MRI image with various noise levels. We compared the performance of our proposed methods with a classical NLM filter using self-weights of 1 [4] and the state-of-the-art LJS method, which has already been shown to be the best among all other previous self-weight determination methods [33].

This article is an extension of a work that was presented at the 2016 IEEE International Symposium on Biomedical Imaging (ISBI) [35], and goes into more depth regarding the theory of the minimax estimator and provides an evaluation of the methods using a significantly larger image dataset.

This paper is organized as follows. Section II reviews the classical NLM filter and revisits the LJS method. Section III investigates the LJS method in terms of the local neighborhood size for self-weight estimation and the potential for excessively large self-weight estimates. Then, Section IV proposes novel LMM-DB and LMM-RP methods using Baranchik's minimax estimator in order to overcome two limitations of the LJS method. Section V illustrates the performance of our proposed methods by providing our simulation results. Lastly, Sections VI and VII discuss and then conclude this paper, respectively.

## II. Review of the Local James-Stein Self-Weight Estimation Method for the NLM Filter

In this section, we will briefly review both the classical NLM method proposed by Buades *et al.* [4] and the LJS self-weight selection method proposed by Wu *et al.* [33].

### A. Reviewing the Classical Non-Local Means Filter

Let us assume that an image $\mathbf{x}$ is contaminated by noise $\mathbf{n}$, which produces a noisy image $\mathbf{y}$:

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

where $\mathbf{n}$ is zero-mean white Gaussian noise with standard deviation $\sigma$. The NLM filtered value at the pixel $i$ is the weighted average of all pixels in a search region $\mathbf{\Omega}_i$:

$$\hat{x}_i = \frac{\sum_{j \in \Omega_i} w_{i,j} y_j}{\sum_{j \in \Omega_i} w_{i,j}} \tag{2}$$

where $y_i$ is the $i$th element of $\mathbf{y}$, $w_{i,j}$ is the weight between the $i$th and $j$th pixels, and $\mathbf{\Omega}_i$ is the set of all pixels in an area around the $i$th pixel, which could be an entire image. The similarity weight of the classical NLM is defined as:

$$w_{i,j} = \exp\left( \frac{-\left\|\mathbf{P}_i\mathbf{y} - \mathbf{P}_j\mathbf{y}\right\|^2}{2\,|\mathbf{P}|\,h^2} \right) \tag{3}$$

where $\mathbf{P}_i$ is an operator used to extract a square-shaped patch centered at the $i$th pixel, $\|\cdot\|$ is an $l_2$ norm, $|\mathbf{P}|$ is the number of pixels within a patch, and $h$ is a global smoothing parameter. Equation (3) implies that the self-weights $w_{i,i}$ are always equal to 1. Previous works on self-weights have shown that good strategies for determining the self-weights also affect the image quality of the NLM filtering [4], [29], [32], [33].

### B. Reviewing Local James-Stein Self-Weight Estimation

The LJS method was proposed in order to determine $w_{i,i}$ as follows [33]. First, (2) was decomposed into two terms:

$$\hat{x}_i = \frac{W_i}{W_i + w_{i,i}} \hat{z}_i + \frac{w_{i,i}}{W_i + w_{i,i}} y_i \tag{4}$$

where $W_i = \sum_{j \in \Omega_i \setminus \{i\}} w_{i,j}$ and

$$\hat{z}_i = \sum_{j \in \Omega_i \setminus \{i\}} w_{i,j} y_j / W_i. \tag{5}$$

The terms $\hat{z}_i$ do not contain $w_{i,i}$. Then, the LJS method reparametrized (4) using

$$p_i = \frac{w_{i,i}}{W_i + w_{i,i}} \tag{6}$$

so that (4) became:

$$\hat{x}_i = (1 - p_i)\,\hat{z}_i + p_i y_i. \tag{7}$$

The problem of estimating the self-weights $w_{i,i}$ became the problem of estimating $p_i$. Lastly, the JS estimator [36], [37] for $p_i$ was proposed:

$$p_i^{\mathrm{LJS}} = 1 - \frac{(|\mathbf{B}| - 2)\,\sigma^2}{\left\| \mathbf{B}_i \mathbf{y} - \mathbf{B}_i \hat{\mathbf{z}} \right\|^2} \tag{8}$$

where $\mathbf{B}_i$ is an operator used to extract a square-shaped neighborhood centered at the $i$th pixel, $|\mathbf{B}|$ is the number of pixels within that neighborhood, and $\sigma$ is the known noise level.

Equation (8) implies that $p_i^{\mathrm{LJS}} \in (-\infty, 1]$. Since the weights are non-negative, it was proposed to use the zero-lower bound for $p_i^{\mathrm{LJS}}$ as follows [33]:

$$\hat{x}_i^{\mathrm{LJS}_+} = \left(1 - p_i^{\mathrm{LJS}_+}\right) \hat{z}_i + p_i^{\mathrm{LJS}_+} y_i \tag{9}$$

where

$$p_i^{\mathrm{LJS}_+} := [p_i^{\mathrm{LJS}}]_+ = \left[ 1 - \frac{(|\mathbf{B}| - 2)\,\sigma^2}{\left\| \mathbf{B}_i \mathbf{y} - \mathbf{B}_i \hat{\mathbf{z}} \right\|^2} \right]_+ \tag{10}$$

and $[s]_+ := \max\ (s, 0)$. Wu *et al.* also mentioned that a user-defined upper bound for $p_i$ can be used, but did not investigate further [33]. It should be noted that the JS estimator does not guarantee its optimality when bounding $p_i^{\mathrm{LJS}}$ in (8).

## III. Limitations of the Local James-Stein Self-Weight Estimation for the NLM Filter

We now investigate two limitations of the original LJS method [33] in terms of the size of local neighborhoods for self-weight estimation, and the potential for excessive self-weight estimation.

### A. Size of Local Neighborhood for Self-Weight Estimation

In the method described in [33], there are two implicit steps required in order to obtain the LJS self-weight estimator (10). The first step is to choose a local set of pixels around the $i$th pixel, referred to as set $\Omega_i^B$, that correspond to the operator $\mathbf{B}_i$, and assume that:

$$\hat{x}_j = (1 - p_i)\,\hat{z}_j + p_i y_j, \quad j \in \Omega_i^B. \tag{11}$$

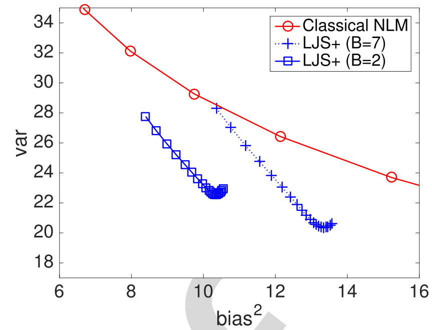



Fig. 1. Bias-variance curves (cameraman example) for the classical NLM and LJS methods (LJS$_+$) for different sizes of local neighborhoods (B). The curves were plotted while varying the smoothing parameter $h$ ($log_2\ h \in [1.8, 3.2]$).

Based on the works of Stein [36] and James and Stein [37], if $|\mathbf{B}| \geq 3$, then for a neighborhood $\Omega_i^B$ extracted using $\mathbf{B}_i$,

$$\hat{x}_j = \left(1 - p_i^{\mathrm{LJS}_+}\right) \hat{z}_j + p_i^{\mathrm{LJS}_+} y_j, \quad j \in \Omega_i^B. \tag{12}$$

is a dominant estimator for $x_j$ "locally" in $\Omega_i^B$. The LJS method used the zero lower bound when estimating $p_i$ in order to obtain a realistic non-negative self-weight value. This was also a good choice in terms of the estimator performance since the positive part of the JS estimator is dominant over the original JS estimator, according to the works of Baranchik [34], [38] and Efron and Morris [39].

The second implicit step is to assign the resulting $p_i^{\mathrm{LJS}}$ to $p_i$ in (7) for only the single pixel $i$ so that:

$$\hat{x}_i^{\mathrm{LJS}} = \left(1 - p_i^{\mathrm{LJS}}\right) \hat{z}_i + p_i^{\mathrm{LJS}} y_i. \tag{13}$$

Wu *et al.* evaluated the LJS method with $|\mathbf{B}| = 15 \times 15$ [33], which seems relatively large.

Based on this implicit two-step interpretation, we can surmise that using a smaller size of $|\mathbf{B}|$ may be more desirable for obtaining a less biased estimate of $p_i$ since the assumption of having the same $p_i$ in $\Omega_i^B$ is less likely to be true for larger sizes of $\Omega_i^B$. Figure 1 confirms our conjecture. The bias-variance curves of the LJS method yielded better bias-variance trade-offs than those in the classical NLM method for both large local neighborhoods with a half window size $B = 7$ ($|\mathbf{B}| = 15 \times 15$) and small local neighborhoods with $B = 2$ ($|\mathbf{B}| = 5 \times 5$). However, using larger local neighborhood sizes for estimating $p_i$ yielded stronger biases than those estimated using smaller sizes for the same level of variance.

### B. Excessively Large Self-Weight Estimates

In the LJS method for determining the self-weights by estimating values for $p_i$ [33], it is theoretically possible that the self-weights have excessively high values. For example, (6) suggests that if $p_i = 1$ and $W_i > 0$, then $w_{i,i} \gg 1$. Slight artifacts were observed in [33] in the background area that were potentially caused by excessive self-weight estimates when a relatively larger neighborhood size $|\mathbf{B}| = 15 \times 15$ was used. We observed a significantly higher degree of degradation in the visual image quality in the background area when the

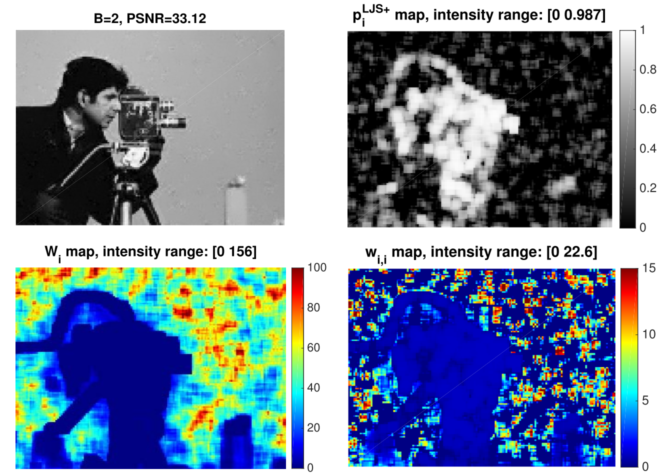



Fig. 2. Denoised image of the cameraman example using the original LJS method [33] with no upper bound for the self-weights (top left), estimated $p_i$ values (top right), calculated $W_i$ (bottom left), and resulting self-weights ($w_{i,i}$) showing excessive self-weights (bottom right). B = 2 and $\sigma = 10$.

size of $|\mathbf{B}|$ in (8) was small, as shown in the image in the top left figure of Fig. 2.

We investigated this issue using an example of the cameraman image that was denoised using the LJS method [33], but with a smaller neighborhood size $|\mathbf{B}| = 5 \times 5$. For areas with more details, such as edges and textures, large $p_i$ values were estimated and yielded large self-weights, as shown in the top right figure of Fig. 2. However, since the values for $W_i$ were also very small in these areas, as shown in the bottom left image in Fig. 2, the resulting self-weight map yielded values close to 1 in the areas with details as shown in the bottom right image in Fig. 2.

In contrast, for areas with almost no details, such as those with a flat intensity background, relatively smaller $p_i$ values were estimated, some of which were much larger than 0 while the rest were closer to 0, as shown in the top right image in Fig. 2. However, since the $W_i$ values for the flat areas were relatively large, as shown in the bottom left image in Fig. 2, some of the estimated $p_i$ values obtained using the LJS method (LJS+) were estimated to yield excessively large self-weights that were much larger than 1, as shown in the bottom right image of Fig. 2. Consequently, these excessively large self-weights caused severe rare patch artifacts in the filtered image, which resulted in visual quality degradation, as observed in the top left image of Fig. 2.

## IV. Local Minimax Estimation Methods for Upper Bounded Self-Weights in a NLM Filter

In this section, we propose two local upper bounded self-weight estimation methods that use Baranchik's minimax estimator [34].

### A. Bounded Self-Weights

It is usually assumed that the self-weights satisfy $w_{i,i} \in [0, 1]$. However, there are many possible upper bounds for the self-weights, including 1 [4] or some positive value that is possibly less than 1 based on SURE [32]. In this article, two different upper bound values $w_{i,i}^{\max}$ for the self-weights were evaluated such that $0 \le w_{i,i} \le w_{i,i}^{\max}$. One upper bound was:

$$w_{i,i}^{\max-\text{one}} = 1, \tag{14}$$

which is the usual choice for the self-weights in the classical NLM method [4]. The other upper bound was:

$$w_{i,i}^{\max-\text{stein}} = \exp\left(-\sigma^2/h^2\right), \tag{15}$$

which was motivated by the SURE-based NLM self-weights [32]. We assume that $\sigma$ is known and $h$ is pre-determined, which means that the upper bound for the self-weights can also be determined in advance. Equation (15) takes the noise level into account. As $\sigma$ is smaller, the maximum self-weight in (15) is closer to one. It should be noted that the difference between (15) and (14) will be greater at higher noise levels.

Since $p_i$ is estimated instead of $w_{i,i}$, it is necessary to derive the range of $p_i$ that corresponds to $0 \le w_{i,i} \le w_{i,i}^{\max}$. From (6), the derivative of $p_i$ with respect to $w_{i,i}$ is non-negative as follows:

$$\frac{d}{dw_{i,i}} p_i = \frac{W_i}{(W_i + w_{i,i})^2} \ge 0$$

since $W_i \ge 0$. Therefore, $p_i$ is a non-decreasing function of $w_{i,i}$ and for $0 \le w_{i,i} \le w_{i,i}^{\max}$, the range of $p_i$ will be

$$0 \le p_i \le \frac{w_{i,i}^{\max}}{W_i + w_{i,i}^{\max}} =: p_i^{\max} \le 1.$$

Note that if $W_i = 0$, then $p_i^{\max} = 1$. The estimator $p_i^{\text{LJS}+}$ in (10) automatically guarantees that $0 \le p_i \le 1$ if $|\mathbf{B}| \ge 2$. However, since $W_i > 0$ generally holds for most real images with noise, it is necessary to constrain $p_i$ to be less than or equal to the upper bound $p_i^{\max}$, which is usually less than one.

### B. Local Minimax Self-Weight Estimation With Direct Bound

Enforcing the upper limit $p_i^{\max}$ on the estimated $p_i$ in (10) using $\min(p_i^{\text{LJS}+}, p_i^{\max})$ breaks the optimality of the JS estimator if $p_i^{\max} < 1$. In this article, we propose using Baranchik's minimax estimator [34] to incorporate bounded self-weights into the estimator (see Baranchik [34], Erfon and Morris [39], and Strawderman [40] for more details on this minimax estimator).

*Theorem 1 (Baranchik):* For $\mathbf{y} \sim \mathcal{N}_r\left(\mathbf{x}, \sigma^2\mathbf{I}\right)$, $r \ge 3$, and loss $L(\mathbf{x}, \hat{\mathbf{x}}) = \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|$, an estimator of the form $\hat{\mathbf{x}} = q\mathbf{y}$ where

$$q = \left[1 - c\left(\|\mathbf{y}\|\right)\frac{\sigma^2(r-2)}{\|\mathbf{y}\|^2}\right] \tag{16}$$

is the minimax, provided that:

(i) $0 \le c\left(\|\mathbf{y}\|\right) \le 2$ and

(ii) the function $c(\cdot)$ is nondecreasing.

Here $\mathbf{y}$ shrinks toward $\mathbf{0}$ which is the initial estimate of $\mathbf{x}$.

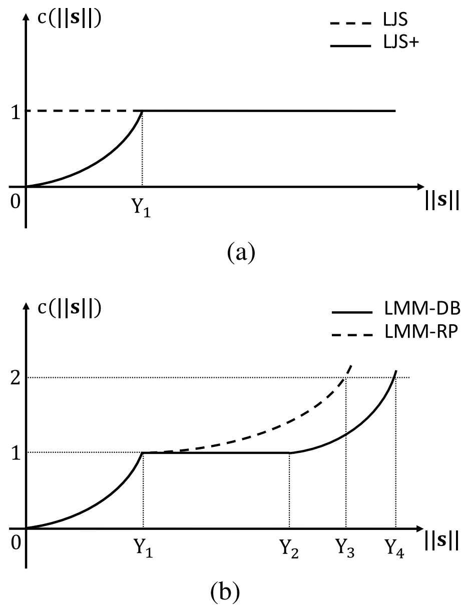



Fig. 3. Graphical illustrations of the original and positive part JS estimators without upper bounds, and the proposed minimax self-weight estimators with upper bounds in terms of $c(\|\mathbf{s}\|)$ vs. $\|\mathbf{s}\|$. (a) Original and positive-part JS estimators. (b) Proposed minimax estimators with bounds.

The original JS estimator and its positive part are special cases of Baranchik's minimax estimator. For the original JS estimator (8):

$$c(\|\mathbf{s}\|) = 1, \tag{17}$$

where $\mathbf{s} = \mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}$ so that both conditions (i, ii) of the Baranchik's theorem are satisfied. In the positive part estimator (9), it can be shown that:

$$c(\|\mathbf{s}\|) = \begin{cases} \dfrac{\|\mathbf{s}\|^2}{\sigma^2(r-2)}, & 0 \le \|\mathbf{s}\| \le \Upsilon_1 \\ 1, & \text{otherwise} \end{cases} \tag{18}$$

where $\Upsilon_1 := \sigma\sqrt{r-2}$. The original and positive part JS estimators are illustrated in Fig. 3 (a).

We propose a new local minimax self-weight estimation method that uses a direct bound with a specific upper-bound value, as follows:

$$p_i^{\text{LMM-DB}} := \min(p_i^{\text{LJS}_+}, p_i^{\max}). \tag{19}$$

This estimator is minimax under certain conditions that can be derived using Baranchik's minimax estimator theorem. According to this theorem, this operation can be interpreted as follows:

$$c(\|\mathbf{s}\|) = \begin{cases} \dfrac{\|\mathbf{s}\|^2}{\sigma^2(r-2)}, & 0 \le \|\mathbf{s}\| \le \Upsilon_1 \\ 1, & \Upsilon_1 < \|\mathbf{s}\| \le \Upsilon_2 \\ \dfrac{\|\mathbf{s}\|^2(1-p^{\max})}{\sigma^2(r-2)}, & \Upsilon_2 < \|\mathbf{s}\| \end{cases} \tag{20}$$

where $\Upsilon_2 := \sigma\sqrt{(r-2)/(1-p^{\max})}$. We call this a local minimax self-weight estimator using direct bound (LMM-DB), which is illustrated in Fig. 3 (b) where $\Upsilon_4 := \sigma\sqrt{2(r-2)/(1-p^{\max})}$.

However, note that LMM-DB is not minimax for $\|\mathbf{s}\| > \Upsilon_4$. Fortunately, $\|\mathbf{s}\| = \|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|$ can be limited by adjusting the smoothing parameter $h$ by making it smaller so that all $\|\mathbf{s}\| \le \Upsilon_4$ and $c(\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|) \le 2$. Then, the LMM-DB becomes "practically" a minimax estimator. Let us denote the maximum $h$ that satisfies $\|\mathbf{s}\| \le \Upsilon_4$ as $h^{\max}$.

In this case, a question can be raised: will the optimal value for $h$ fall into the range of $h$ that satisfies $\|\mathbf{s}\| \le \Upsilon_4$? Interestingly, our simulations with many natural images showed that the optimal smoothing parameter $h^*$ based on the true images is very close to $h^{\max}$. This is because the LMM-DB yielded $p^{\max} \to 1$ so that $\Upsilon_2 \to \infty$, and almost all $\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|$ were less than or equal to $\Upsilon_4$. Therefore, $p_i^{\text{LMM-DB}}$ is "practically" a minimax value based on Baranchik's theorem for many natural images. Moreover, the LMM-DB method may provide a way to choose the optimal global smoothing parameter value $h$ without knowing the underlying true image. We empirically investigate this issue in Section V.

### *C. Local Minimax Self-Weight Estimation With Re-Parametrization*

The LMM-DB algorithm set $p$ to be the same $p^{\max}$ for a wide range of $\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|$ values. We now propose another new method, called the local minimax self-weight estimation with reparametrization (LMM-RP) method, that assigns different $p$ values for different $\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|$.

We reparametrized $p_i$ in (7) in the following way:

$$\begin{aligned} \hat{x}_i &= \hat{z}_i(p_i/p_i^{\max})p_i^{\max}(y_i - \hat{z}_i) \\ &= \hat{z}_i + p_i^{\mathrm{T}}(y_i^{\mathrm{T}} - \hat{z}_i^{\mathrm{T}}) \qquad (21) \\ &= (1 - p_i^{\max})\hat{z}_i + \hat{z}_i^{\mathrm{T}} + p_i^{\mathrm{T}}(y_i^{\mathrm{T}} - \hat{z}_i^{\mathrm{T}}) \qquad (22) \end{aligned}$$

where $\hat{z}_i^{\mathrm{T}} = p_i^{\max}\hat{z}_i$, $y_i^{\mathrm{T}} = p_i^{\max}y_i$, and

$$p_i^{\mathrm{T}} = \frac{1}{p_i^{\max}}\frac{w_{i,i}}{W_i + w_{i,i}}. \tag{23}$$

Note that for $0 \le w_{i,i} \le w_{i,i}^{\max}$, $p_i^{\mathrm{T}}$ is an increasing function of $w_{i,i}$ and the range of $p_i^{\mathrm{T}}$ is $0 \le p_i^{\mathrm{T}} \le 1$. We propose to use the positive part of the JS estimator to estimate the reparametrized $p_i^{\mathrm{T}}$, as follows:

$$\begin{aligned} p_i^{\mathrm{T,LJS}_+} &= \left[1 - \frac{(|\mathbf{B}| - 2)\,(p_i^{\max})^2\sigma^2}{\|\mathbf{B}_i\mathbf{y}^{\mathrm{T}} - \mathbf{B}_i\hat{\mathbf{z}}^{\mathrm{T}}\|^2}\right]_+ \\ &= \left[1 - \frac{(|\mathbf{B}| - 2)\,\sigma^2}{\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|^2}\right]_+ = p_i^{\mathrm{LJS}_+}. \qquad (24) \end{aligned}$$

This method is equivalent to using a multiplicative factor $p_i^{\max}$ for the original JS shrinkage (9):

$$\hat{x}_i^{\text{LMM-RP}} = (1 - p_i^{\text{LMM-RP}})\hat{z}_i + p_i^{\text{LMM-RP}}y_i \tag{25}$$

where

$$p_i^{\text{LMM-RP}} = p_i^{\max}\left[1 - \frac{(|\mathbf{B}| - 2)\,\sigma^2}{\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|^2}\right]_+. \tag{26}$$

This proposed LMM-RP estimator is not dominant when estimating $x_i$, but rather is dominant when estimating $p_i^{\max}x_i$, as shown in (22). Thus, the positive part JS estimator does not guarantee that the LMM-RP is dominant.

Baranchik's minimax estimation theorem can be used to analyze the LMM-RP estimator as follows:

$$c\left(\|\mathbf{s}\|\right) = \begin{cases} \dfrac{\|\mathbf{s}\|^2}{\sigma^2(r-2)}, & 0 \le \|\mathbf{s}\| \le \Upsilon_1 \\ \dfrac{\|\mathbf{s}\|^2(1-p^{\max})}{\sigma^2(r-2)} + p^{\max}, & \Upsilon_1 < \|\mathbf{s}\| \end{cases} \tag{27}$$

where if $\|\mathbf{s}\|$ is $\Upsilon_3 := \sigma\sqrt{(2-p^{\max})(r-2)/(1-p^{\max})}$, then $c\left(\|\mathbf{s}\|\right) = 2$. The LMM-RP method is also illustrated in Fig. 3 (b), and is minimax if $\|\mathbf{s}\| \le \Upsilon_3$. The global smoothing parameter $h$ can be adjusted so that this condition is satisfied for different images. As in the case of the LMM-DB, it turns out that the optimal global smoothing parameter $h^*$ and the upper bound $h$ that satisfies $\|\mathbf{s}\| \le \Upsilon_3$ are also very close to each other when the LMM-RP method is applied to many natural images. Therefore, the LMM-RP method is "practically" a minimax. The following table summarizes the LJS self-weight estimation method and our proposed LMM-based self-weight estimation methods.

**Summary of Self-Weight Estimation Methods**

$\mathbf{LJS_+}$ [33] :

$$p_i^{\mathrm{LJS}_+} = \left[1 - (|\mathbf{B}|-2)\,\sigma^2 / \left\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\right\|^2\right]_+$$

$$\hat{x}_i^{\mathrm{LJS}_+} = (1 - p_i^{\mathrm{LJS}_+})\hat{z}_i + p_i^{\mathrm{LJS}_+} y_i$$

$\mathbf{LMM-DB}$ :

$$p_i^{\mathrm{LMM-DB}} = \min(p_i^{\mathrm{LJS}_+}, p_i^{\max})$$

$$\hat{x}_i^{\mathrm{LMM-DB}} = (1 - p_i^{\mathrm{LMM-DB}})\hat{z}_i + p_i^{\mathrm{LMM-DB}} y_i$$

$\mathbf{LMM-RP}$ :

$$p_i^{\mathrm{LMM-RP}} = p_i^{\mathrm{LJS}_+} p_i^{\max}$$

$$\hat{x}_i^{\mathrm{LMM-RP}} = (1 - p_i^{\mathrm{LMM-RP}})\hat{z}_i + p_i^{\mathrm{LMM-RP}} y_i$$

## V. Simulation Results

### A. Simulation Setup

Ten natural images[1] (cameraman, lena, montage, house, pepper, barbara, boat, hill, couple, fingerprint) and five images from the SUN database[2] (abbey, airplane cabin, airport terminal, alley, amphitheater) were used in our study as noise-free images (128 × 128, 256 × 256, or 512 × 512 pixels, 8 bits). A real patient MRI (512 × 512 pixels, 8 bits) that was acquired and processed under institutional review board (IRB) approved protocols was also used. White Gaussian noise was added to each input image with various standard deviations $\sigma \in \{10, 20, 40, 60\}$.

All algorithms were implemented using MATLAB R2015b (The Mathworks, Inc., Natick, MA, USA). The patch size and search window size of the NLM filter were chosen to be 7 × 7 and 31 × 31, respectively, which were the same as those used in [33]. Both the state-of-the-art LJS algorithm and the proposed algorithms were tested using $B = 1, \cdots, 9$ where $|\mathbf{B}| = (2B+1)^2 > 3$.

The global smoothing parameter $h$ was chosen empirically to yield the best PSNR:

$$\mathrm{PSNR}\left(\hat{\mathbf{x}}\right) = 10\log_{10}\frac{255^2}{\left\|\hat{\mathbf{x}} - \mathbf{x}\right\|^2/N}, \tag{28}$$

where $N$ is the size of the image. In addition to the PSNR, the mean bias vs. the mean variance trade-off curves were used as performance measures for the different smoothing parameter values $h$:

$$\overline{\mathrm{bias}^2} = \frac{1}{N}\sum_{i=1}^{N}(\bar{x}_i - x_i)^2, \tag{29}$$

$$\overline{\mathrm{var}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{k-1}\sum_{j=1}^{k}\left(\hat{x}_{ij} - \bar{x}_i\right)^2, \tag{30}$$

where $k$ is the number of realizations (20 in our simulation), $\hat{x}_{ij}$ is the $j$th estimation at the $i$th pixel, and $\bar{x}_i$ is the mean of $\hat{x}_{ij}$, as given by:

$$\bar{x}_i = \frac{1}{k}\sum_{j=1}^{k}\hat{x}_{ij}.$$

A visual quality assessment was also performed.

### B. Performance Studies Using the PSNR

In order to estimate values of $p_i$ for a fixed neighborhood size $B$, the optimal NLM smoothing parameter $h^*$ was determined such that the PSNR was maximized. In our proposed methods, the two maximum self-weights in (14) and (15) were used. The LMM-DB and LMM-RP methods given by (14) are denoted $\mathrm{LMM-DB^{one}}$ and $\mathrm{LMM-RP^{one}}$, while the LMM-DB and LMM-RP methods given by (15) are denoted $\mathrm{LMM-DB^{stein}}$ and $\mathrm{LMM-RP^{stein}}$. Table I summarizes the quantitative PSNR results for the 16 images with 4 different noise levels. When $B = 7$, our proposed LMM-DB and LMM-RP methods based on Baranchik's minimax estimator yielded much better PSNR results than did setting the self-weight to one in the classical NLM method [4], and comparable PSNR values to the LJS method based on the JS estimator [33]. When $B = 2$, our proposed LMM-DB and LMM-RP methods yielded better PSNR values than did the LJS.

For the five examples of lena, house, peppers, barbara, boat with $\sigma = 20$, PSNRs of our proposed methods (global smoothing parameter and fixed neighborhoods, but adaptive self-weight) were 0.72 ∼ 0.97 dB better than classical NLM. In [10], it is reported that for the same five examples with the same level of noise, the work of Kervrann et al. (fixed self-weight, but local smoothing parameters and adaptive neighborhoods) yielded 0.99 ∼ 1.55 dB better PSNR than classical NLM. Self-weights, local smoothing parameter, neighborhoods size are important factors in the NLM filter to determine output image quality.

[1] Available online at: http://www.cs.tut.fi/~foi/GCF-BM3D/BM3D_images.zip as the date of 16 Nov. 2015.

[2] Available online at: http://vision.princeton.edu/projects/2010/SUN/ as the date of 16 Sep. 2016.

TABLE I

PSNR (dB) SUMMARY (MEAN ± STANDARD DEVIATION) FOR VARIOUS NATURAL IMAGES

| | | | B = 2 | | | | | B = 7 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | σ | Classical NLM | LJS+ | LMM-DB one | LMM-RP one | LMM-DB stein | LMM-RP stein | LJS+ | LMM-DB one | LMM-RP one | LMM-DB stein | LMM-RP stein |
| ***cameraman*** | *10* | 32.42 ± 0.034 | 33.12 ± 0.031 | **33.32 ± 0.030** | 33.29 ± 0.029 | 33.17 ± 0.029 | 33.04 ± 0.030 | 32.98 ± 0.035 | 33.10 ± 0.036 | 33.05 ± 0.037 | 32.98 ± 0.038 | 32.85 ± 0.040 |
| | *20* | 28.48 ± 0.052 | 29.12 ± 0.052 | **29.46 ± 0.056** | 29.29 ± 0.056 | 29.27 ± 0.060 | 28.97 ± 0.052 | 29.32 ± 0.062 | 29.34 ± 0.059 | 29.04 ± 0.054 | 29.11 ± 0.058 | 28.80 ± 0.049 |
| | *40* | 25.35 ± 0.059 | 25.39 ± 0.083 | 25.89 ± 0.075 | **26.11 ± 0.082** | 26.08 ± 0.087 | 25.74 ± 0.079 | 25.98 ± 0.073 | 25.98 ± 0.073 | 25.94 ± 0.080 | 25.96 ± 0.083 | 25.63 ± 0.076 |
| | *60* | 23.19 ± 0.065 | 22.88 ± 0.062 | 23.39 ± 0.055 | **23.74 ± 0.065** | 23.67 ± 0.069 | 23.54 ± 0.061 | 23.68 ± 0.068 | 23.69 ± 0.068 | 23.62 ± 0.071 | 23.63 ± 0.070 | 23.54 ± 0.059 |
| ***lena*** | *10* | 33.90 ± 0.018 | 34.52 ± 0.017 | 34.74 ± 0.017 | 34.81 ± 0.018 | 34.77 ± 0.017 | 34.69 ± 0.017 | 34.82 ± 0.020 | **34.83 ± 0.020** | 34.80 ± 0.019 | 34.76 ± 0.018 | 34.63 ± 0.018 |
| | *20* | 30.78 ± 0.031 | 30.90 ± 0.023 | 31.25 ± 0.033 | 31.50 ± 0.032 | **31.51 ± 0.032** | 31.31 ± 0.029 | **31.51 ± 0.031** | **31.51 ± 0.031** | 31.45 ± 0.028 | 31.48 ± 0.031 | 31.27 ± 0.029 |
| | *40* | 27.64 ± 0.032 | 26.94 ± 0.028 | 27.74 ± 0.033 | 28.08 ± 0.029 | 28.08 ± 0.029 | 28.06 ± 0.028 | **28.10 ± 0.028** | **28.10 ± 0.028** | 28.07 ± 0.028 | 28.08 ± 0.029 | 28.06 ± 0.028 |
| | *60* | 25.60 ± 0.052 | 24.38 ± 0.040 | 25.66 ± 0.051 | **26.02 ± 0.052** | 26.01 ± 0.053 | **26.02 ± 0.054** | 26.01 ± 0.051 | 26.01 ± 0.051 | **26.02 ± 0.054** | **26.02 ± 0.053** | **26.02 ± 0.054** |
| ***montage*** | *10* | 34.68 ± 0.045 | 35.19 ± 0.043 | 35.60 ± 0.042 | **35.67 ± 0.039** | 35.65 ± 0.042 | 35.55 ± 0.045 | 35.12 ± 0.049 | 35.39 ± 0.046 | 35.38 ± 0.042 | 35.46 ± 0.045 | 35.34 ± 0.047 |
| | *20* | 30.35 ± 0.088 | 30.74 ± 0.062 | 31.29 ± 0.067 | 31.38 ± 0.070 | **31.40 ± 0.078** | 31.06 ± 0.068 | 31.00 ± 0.076 | 31.13 ± 0.073 | 31.07 ± 0.068 | 31.18 ± 0.068 | 30.81 ± 0.063 |
| | *40* | 26.24 ± 0.063 | 26.30 ± 0.072 | 26.98 ± 0.070 | 27.29 ± 0.063 | **27.30 ± 0.064** | 27.20 ± 0.061 | 27.01 ± 0.052 | 27.03 ± 0.052 | 27.04 ± 0.055 | 27.08 ± 0.053 | 27.00 ± 0.053 |
| | *60* | 23.76 ± 0.104 | 23.48 ± 0.116 | 24.16 ± 0.113 | **24.61 ± 0.115** | 24.60 ± 0.106 | 24.38 ± 0.115 | 24.33 ± 0.092 | 24.33 ± 0.092 | 24.36 ± 0.090 | 24.37 ± 0.088 | 24.16 ± 0.095 |
| ***house*** | *10* | 34.57 ± 0.038 | 35.02 ± 0.039 | 35.36 ± 0.041 | **35.38 ± 0.039** | 35.34 ± 0.043 | 35.29 ± 0.046 | 35.31 ± 0.042 | 35.32 ± 0.043 | 35.25 ± 0.044 | 35.21 ± 0.047 | 35.12 ± 0.045 |
| | *20* | 31.43 ± 0.063 | 31.54 ± 0.048 | 32.13 ± 0.050 | **32.39 ± 0.048** | **32.39 ± 0.050** | 32.19 ± 0.067 | 32.30 ± 0.052 | 32.31 ± 0.054 | 32.26 ± 0.058 | 32.30 ± 0.056 | 32.10 ± 0.068 |
| | *40* | 27.62 ± 0.044 | 27.18 ± 0.038 | 27.84 ± 0.049 | **28.37 ± 0.037** | **28.37 ± 0.039** | 28.33 ± 0.045 | 28.35 ± 0.041 | 28.35 ± 0.041 | 28.34 ± 0.045 | 28.35 ± 0.042 | 28.33 ± 0.045 |
| | *60* | 25.01 ± 0.092 | 24.24 ± 0.087 | 25.17 ± 0.098 | 25.65 ± 0.095 | 25.65 ± 0.087 | 25.65 ± 0.088 | 25.65 ± 0.088 | 25.65 ± 0.088 | 25.65 ± 0.087 | **25.66 ± 0.087** | 25.65 ± 0.088 |
| ***peppers*** | *10* | 32.62 ± 0.056 | 33.37 ± 0.042 | 33.53 ± 0.048 | **33.56 ± 0.049** | 33.49 ± 0.051 | 33.39 ± 0.050 | 33.28 ± 0.043 | 33.37 ± 0.040 | 33.35 ± 0.042 | 33.33 ± 0.042 | 33.17 ± 0.042 |
| | *20* | 28.94 ± 0.031 | 29.54 ± 0.029 | 29.78 ± 0.040 | **29.88 ± 0.038** | 29.86 ± 0.028 | 29.51 ± 0.027 | 29.77 ± 0.027 | 29.79 ± 0.027 | 29.70 ± 0.026 | 29.73 ± 0.024 | 29.34 ± 0.033 |
| | *40* | 25.31 ± 0.050 | 25.50 ± 0.057 | 25.67 ± 0.041 | **26.12 ± 0.049** | 26.11 ± 0.054 | 25.97 ± 0.055 | 26.08 ± 0.054 | 26.08 ± 0.054 | 26.04 ± 0.056 | 26.05 ± 0.054 | 25.95 ± 0.055 |
| | *60* | 22.99 ± 0.048 | 22.95 ± 0.091 | 23.18 ± 0.061 | 23.80 ± 0.067 | 23.80 ± 0.071 | 23.78 ± 0.075 | **23.81 ± 0.070** | **23.81 ± 0.070** | 23.79 ± 0.074 | 23.80 ± 0.073 | 23.78 ± 0.075 |
| ***barbara*** | *10* | 32.93 ± 0.026 | 33.50 ± 0.018 | 33.66 ± 0.020 | 33.70 ± 0.021 | 33.66 ± 0.022 | 33.53 ± 0.020 | 33.72 ± 0.017 | **33.74 ± 0.017** | 33.69 ± 0.017 | 33.66 ± 0.017 | 33.44 ± 0.017 |
| | *20* | 29.36 ± 0.032 | 29.83 ± 0.029 | 29.96 ± 0.032 | 30.23 ± 0.030 | **30.27 ± 0.028** | 30.04 ± 0.029 | **30.27 ± 0.029** | **30.27 ± 0.028** | 30.19 ± 0.026 | 30.24 ± 0.027 | 30.00 ± 0.030 |
| | *40* | 25.68 ± 0.047 | 25.78 ± 0.048 | 25.79 ± 0.047 | 26.46 ± 0.043 | 26.51 ± 0.040 | 26.51 ± 0.039 | **26.52 ± 0.040** | **26.52 ± 0.040** | 26.51 ± 0.039 | 26.51 ± 0.040 | 26.51 ± 0.039 |
| | *60* | 23.50 ± 0.032 | 23.17 ± 0.039 | 23.57 ± 0.034 | 24.13 ± 0.037 | 24.15 ± 0.035 | **24.16 ± 0.035** | 24.15 ± 0.036 | 24.15 ± 0.036 | **24.16 ± 0.035** | **24.16 ± 0.035** | **24.16 ± 0.035** |
| ***boat*** | *10* | 31.78 ± 0.015 | 32.73 ± 0.019 | 32.81 ± 0.018 | **32.82 ± 0.017** | 32.72 ± 0.015 | 32.61 ± 0.016 | 32.73 ± 0.018 | 32.75 ± 0.018 | 32.72 ± 0.017 | 32.65 ± 0.017 | 32.49 ± 0.017 |
| | *20* | 28.40 ± 0.017 | 29.14 ± 0.017 | 29.23 ± 0.019 | **29.37 ± 0.015** | 29.34 ± 0.015 | 29.05 ± 0.015 | 29.30 ± 0.018 | 29.30 ± 0.018 | 29.25 ± 0.017 | 29.27 ± 0.017 | 28.95 ± 0.018 |
| | *40* | 21.95 ± 0.053 | 25.45 ± 0.021 | 25.45 ± 0.021 | **26.01 ± 0.016** | 25.99 ± 0.016 | 25.92 ± 0.014 | 25.98 ± 0.012 | 25.98 ± 0.012 | 25.95 ± 0.013 | 25.96 ± 0.012 | 25.92 ± 0.014 |
| | *60* | 23.64 ± 0.025 | 23.11 ± 0.028 | 23.72 ± 0.026 | **24.01 ± 0.026** | **24.01 ± 0.025** | 23.99 ± 0.025 | **24.01 ± 0.025** | **24.01 ± 0.025** | 24.00 ± 0.025 | 24.00 ± 0.025 | 23.99 ± 0.025 |
| ***hill*** | *10* | 31.87 ± 0.029 | 32.63 ± 0.020 | 32.67 ± 0.019 | **32.71 ± 0.018** | 32.61 ± 0.018 | 32.47 ± 0.014 | 32.67 ± 0.016 | 32.67 ± 0.016 | 32.64 ± 0.016 | 32.55 ± 0.015 | 32.34 ± 0.013 |
| | *20* | 28.82 ± 0.022 | 29.23 ± 0.031 | 29.23 ± 0.031 | **29.48 ± 0.024** | **29.48 ± 0.023** | 29.29 ± 0.023 | 29.45 ± 0.021 | 29.45 ± 0.021 | 29.41 ± 0.023 | 29.42 ± 0.022 | 29.25 ± 0.024 |
| | *40* | 25.91 ± 0.022 | 25.70 ± 0.026 | 25.98 ± 0.024 | 26.36 ± 0.024 | **26.38 ± 0.022** | 26.37 ± 0.022 | **26.38 ± 0.022** | **26.38 ± 0.022** | 26.37 ± 0.022 | 26.38 ± 0.022 | 26.37 ± 0.022 |
| | *60* | 24.25 ± 0.017 | 23.45 ± 0.013 | 24.32 ± 0.018 | **24.60 ± 0.024** | 24.59 ± 0.024 | **24.60 ± 0.024** | 24.59 ± 0.022 | 24.59 ± 0.022 | **24.60 ± 0.024** | **24.60 ± 0.024** | **24.60 ± 0.024** |
| ***couple*** | *10* | 31.80 ± 0.014 | 32.76 ± 0.009 | 32.81 ± 0.009 | **32.85 ± 0.010** | 32.80 ± 0.011 | 32.71 ± 0.011 | 32.76 ± 0.013 | 32.77 ± 0.013 | 32.75 ± 0.012 | 32.72 ± 0.011 | 32.59 ± 0.012 |
| | *20* | 28.14 ± 0.023 | 28.93 ± 0.023 | 28.93 ± 0.023 | **29.16 ± 0.028** | **29.16 ± 0.029** | 28.86 ± 0.030 | 29.11 ± 0.028 | 29.11 ± 0.028 | 29.07 ± 0.029 | 29.08 ± 0.029 | 28.76 ± 0.030 |
| | *40* | 24.93 ± 0.035 | 25.03 ± 0.026 | 25.05 ± 0.026 | 25.49 ± 0.026 | **25.50 ± 0.030** | 25.44 ± 0.032 | 25.48 ± 0.031 | 25.48 ± 0.031 | 25.47 ± 0.030 | 25.47 ± 0.030 | 25.43 ± 0.032 |
| | *60* | 23.25 ± 0.037 | 22.76 ± 0.043 | 23.29 ± 0.036 | 23.59 ± 0.045 | **23.60 ± 0.044** | 23.59 ± 0.044 | **23.60 ± 0.044** | **23.60 ± 0.044** | 23.59 ± 0.044 | **23.60 ± 0.044** | 23.59 ± 0.044 |
| ***fingerprint*** | *10* | 30.27 ± 0.017 | 30.87 ± 0.015 | 30.87 ± 0.016 | 30.84 ± 0.016 | 30.80 ± 0.017 | 30.57 ± 0.016 | **30.88 ± 0.018** | **30.88 ± 0.019** | 30.83 ± 0.019 | 30.81 ± 0.020 | 30.50 ± 0.018 |
| | *20* | 26.64 ± 0.010 | 27.06 ± 0.014 | 27.06 ± 0.014 | 27.04 ± 0.014 | **27.12 ± 0.012** | 26.72 ± 0.013 | 27.10 ± 0.012 | 27.10 ± 0.012 | 26.93 ± 0.013 | 27.05 ± 0.013 | 26.70 ± 0.012 |
| | *40* | 23.20 ± 0.018 | 23.68 ± 0.024 | 23.68 ± 0.024 | 23.96 ± 0.023 | **24.06 ± 0.022** | 24.05 ± 0.023 | 24.05 ± 0.022 | 24.05 ± 0.022 | 24.05 ± 0.022 | 24.05 ± 0.022 | 24.05 ± 0.023 |
| | *60* | 20.93 ± 0.034 | 21.44 ± 0.029 | 21.44 ± 0.029 | 21.85 ± 0.041 | 21.97 ± 0.037 | **21.98 ± 0.037** | **21.98 ± 0.037** | **21.98 ± 0.037** | **21.98 ± 0.037** | **21.98 ± 0.037** | **21.98 ± 0.037** |
| ***MRI*** | *10* | 40.06 ± 0.043 | 39.19 ± 0.040 | 40.81 ± 0.033 | **40.89 ± 0.032** | 40.83 ± 0.034 | 40.71 ± 0.029 | 40.79 ± 0.040 | 40.85 ± 0.038 | 40.83 ± 0.037 | 40.81 ± 0.038 | 40.61 ± 0.032 |
| | *20* | 36.14 ± 0.067 | 34.47 ± 0.047 | 36.57 ± 0.062 | 36.70 ± 0.063 | 36.74 ± 0.065 | 36.60 ± 0.068 | 36.74 ± 0.063 | **36.77 ± 0.064** | 36.64 ± 0.068 | 36.72 ± 0.066 | 36.59 ± 0.067 |
| | *40* | 32.22 ± 0.067 | 29.33 ± 0.055 | 32.31 ± 0.071 | 32.53 ± 0.069 | 32.53 ± 0.069 | 32.52 ± 0.069 | 32.49 ± 0.064 | **32.54 ± 0.070** | 32.52 ± 0.069 | **32.54 ± 0.069** | 32.52 ± 0.069 |
| | *60* | 29.57 ± 0.056 | 26.13 ± 0.058 | 29.68 ± 0.057 | **29.88 ± 0.056** | **29.88 ± 0.056** | **29.88 ± 0.056** | 29.76 ± 0.060 | 29.86 ± 0.059 | **29.88 ± 0.056** | **29.88 ± 0.056** | **29.88 ± 0.056** |
| ***abbey*** | *10* | 29.31 ± 0.035 | **29.96 ± 0.029** | **29.96 ± 0.029** | 29.86 ± 0.031 | 29.87 ± 0.033 | 29.38 ± 0.030 | 29.92 ± 0.028 | 29.92 ± 0.028 | 29.83 ± 0.032 | 29.83 ± 0.033 | 29.34 ± 0.030 |
| | *20* | 25.53 ± 0.034 | **25.91 ± 0.036** | **25.91 ± 0.036** | 25.87 ± 0.033 | 25.80 ± 0.034 | 25.27 ± 0.034 | 25.87 ± 0.034 | 25.87 ± 0.034 | 25.72 ± 0.032 | 25.71 ± 0.034 | 25.16 ± 0.033 |
| | *40* | 22.85 ± 0.036 | 22.94 ± 0.031 | 22.94 ± 0.031 | 23.10 ± 0.030 | **23.14 ± 0.024** | 23.10 ± 0.024 | 23.13 ± 0.025 | 23.13 ± 0.025 | 23.12 ± 0.025 | 23.12 ± 0.025 | 23.10 ± 0.025 |
| | *60* | 21.60 ± 0.034 | 21.30 ± 0.031 | 21.58 ± 0.032 | 21.83 ± 0.038 | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** |
| ***airplane cabin*** | *10* | 31.42 ± 0.077 | 32.47 ± 0.088 | **32.54 ± 0.086** | 32.49 ± 0.085 | 32.50 ± 0.084 | 32.05 ± 0.082 | 32.36 ± 0.078 | 32.39 ± 0.077 | 32.30 ± 0.085 | 32.33 ± 0.084 | 31.78 ± 0.082 |
| | *20* | 27.52 ± 0.107 | 28.53 ± 0.110 | 28.60 ± 0.117 | **28.70 ± 0.130** | 28.58 ± 0.129 | 28.21 ± 0.119 | 28.57 ± 0.133 | 28.57 ± 0.133 | 28.45 ± 0.127 | 28.46 ± 0.127 | 28.11 ± 0.121 |
| | *40* | 24.62 ± 0.112 | 24.86 ± 0.124 | 24.91 ± 0.105 | 25.26 ± 0.113 | **25.28 ± 0.112** | 25.24 ± 0.121 | 25.25 ± 0.120 | 25.25 ± 0.120 | 25.25 ± 0.118 | 25.25 ± 0.117 | 25.24 ± 0.120 |
| | *60* | 22.89 ± 0.185 | 22.60 ± 0.176 | 22.97 ± 0.194 | 23.36 ± 0.206 | **23.39 ± 0.192** | 23.38 ± 0.193 | 23.38 ± 0.193 | 23.38 ± 0.193 | 23.38 ± 0.193 | 23.38 ± 0.193 | 23.38 ± 0.193 |
| ***airport terminal*** | *10* | 32.79 ± 0.042 | 33.41 ± 0.038 | **33.58 ± 0.036** | 33.51 ± 0.038 | **33.58 ± 0.043** | 33.09 ± 0.045 | 33.26 ± 0.041 | 33.34 ± 0.041 | 33.23 ± 0.044 | 33.37 ± 0.044 | 32.86 ± 0.040 |
| | *20* | 28.74 ± 0.071 | 29.58 ± 0.050 | 29.72 ± 0.044 | 29.95 ± 0.055 | **29.97 ± 0.054** | 29.67 ± 0.049 | 29.70 ± 0.054 | 29.70 ± 0.054 | 29.69 ± 0.058 | 29.74 ± 0.057 | 29.42 ± 0.047 |
| | *40* | 24.78 ± 0.060 | 25.21 ± 0.055 | 25.21 ± 0.055 | 25.70 ± 0.077 | **25.71 ± 0.077** | 25.55 ± 0.070 | 25.61 ± 0.074 | 25.61 ± 0.074 | 25.59 ± 0.074 | 25.60 ± 0.074 | 25.51 ± 0.076 |
| | *60* | 22.72 ± 0.061 | 22.59 ± 0.049 | 22.80 ± 0.063 | 23.22 ± 0.072 | **23.23 ± 0.073** | 23.20 ± 0.070 | 23.21 ± 0.070 | 23.21 ± 0.070 | 23.21 ± 0.071 | 23.21 ± 0.071 | 23.20 ± 0.070 |
| ***alley*** | *10* | 37.58 ± 0.075 | 37.65 ± 0.053 | 37.90 ± 0.070 | 38.44 ± 0.077 | **38.48 ± 0.076** | 38.42 ± 0.082 | 38.44 ± 0.079 | 38.44 ± 0.079 | 38.44 ± 0.079 | 38.45 ± 0.078 | 38.41 ± 0.081 |
| | *20* | 33.90 ± 0.087 | 33.35 ± 0.087 | 34.15 ± 0.102 | 34.59 ± 0.101 | 34.59 ± 0.101 | 34.59 ± 0.101 | 34.59 ± 0.101 | 34.59 ± 0.101 | **34.60 ± 0.101** | **34.60 ± 0.101** | 34.59 ± 0.102 |
| | *40* | 31.06 ± 0.056 | 28.67 ± 0.123 | 31.12 ± 0.057 | **31.27 ± 0.058** | 31.26 ± 0.059 | 31.26 ± 0.060 | 31.16 ± 0.077 | 31.22 ± 0.064 | 31.26 ± 0.059 | **31.27 ± 0.060** | 31.26 ± 0.060 |
| | *60* | 29.65 ± 0.101 | 25.84 ± 0.110 | 29.68 ± 0.101 | **29.77 ± 0.102** | 29.76 ± 0.102 | **29.77 ± 0.102** | 29.46 ± 0.116 | 29.72 ± 0.102 | **29.77 ± 0.102** | **29.77 ± 0.103** | **29.77 ± 0.102** |
| ***amphitheater*** | *10* | 32.27 ± 0.038 | 32.87 ± 0.038 | **33.02 ± 0.034** | 32.94 ± 0.035 | 32.90 ± 0.036 | 32.39 ± 0.050 | 32.94 ± 0.040 | 32.96 ± 0.037 | 32.83 ± 0.052 | 32.81 ± 0.049 | 32.23 ± 0.069 |
| | *20* | 28.70 ± 0.089 | 28.94 ± 0.061 | 29.09 ± 0.064 | **29.12 ± 0.071** | 29.02 ± 0.079 | 28.58 ± 0.078 | 29.11 ± 0.073 | 29.11 ± 0.073 | 28.94 ± 0.081 | 28.91 ± 0.079 | 28.52 ± 0.077 |
| | *40* | 25.88 ± 0.135 | 25.40 ± 0.093 | 25.90 ± 0.126 | 26.02 ± 0.157 | 26.02 ± 0.173 | 25.99 ± 0.175 | **26.06 ± 0.158** | **26.06 ± 0.158** | 26.01 ± 0.173 | 26.01 ± 0.172 | 25.99 ± 0.175 |
| | *60* | 24.62 ± 0.077 | 23.45 ± 0.159 | 24.64 ± 0.080 | 24.72 ± 0.081 | 24.72 ± 0.082 | 24.72 ± 0.082 | 24.72 ± 0.096 | **24.73 ± 0.093** | 24.72 ± 0.082 | 24.72 ± 0.082 | 24.72 ± 0.082 |

### *C. Performance Studies With Bias-Variance Trade-Off*

The bias-variance trade-off was investigated using many natural images. As shown in Fig. 1, a neighborhood size $B$ was used to estimate $p_i$ using the LJS method [33], and this was a significant factor when determining the bias. This tendency was also observed for the other different natural images, as illustrated in Fig. 4. Increasing $B$ in the LJS method moved the bias-variance trade-off curves in the bottom right direction, meaning that the bias increased and the variance decreased. However, the role of the smoothing parameter $h$ changed in the LJS method. Unlike in classical NLM method (see the NLM bias-variance curve in Fig. 1), increasing the smoothing parameter $h$ beyond a certain point in the LJS method did not further decrease the variance in any of the natural images that we tested. This is because increasing $h$ will also increases the $p_i$ values so that the resulting LJS estimator becomes closer to the noisy input image $y_i$ due to the lack of bounds for the self-weights.

Our proposed methods (LMM-DB, LMM-RP) yielded trade-off curves that have decreased variances for increasing values of the smoothing parameter $h$. Figure 5 shows the trade-off curves for the cameraman example for different methods (LMM-DB, LMM-RP), different neighborhood sizes ($B = 2, 7$), and different noise levels ($\sigma = 10, 40$). Our proposed methods yielded bias-variance curves that were less than or equal to those in the LJS method for fixed $B$ and $\sigma$. This tendency was also observed with other natural images, as illustrated in Fig. 4. It was important to choose appropriate

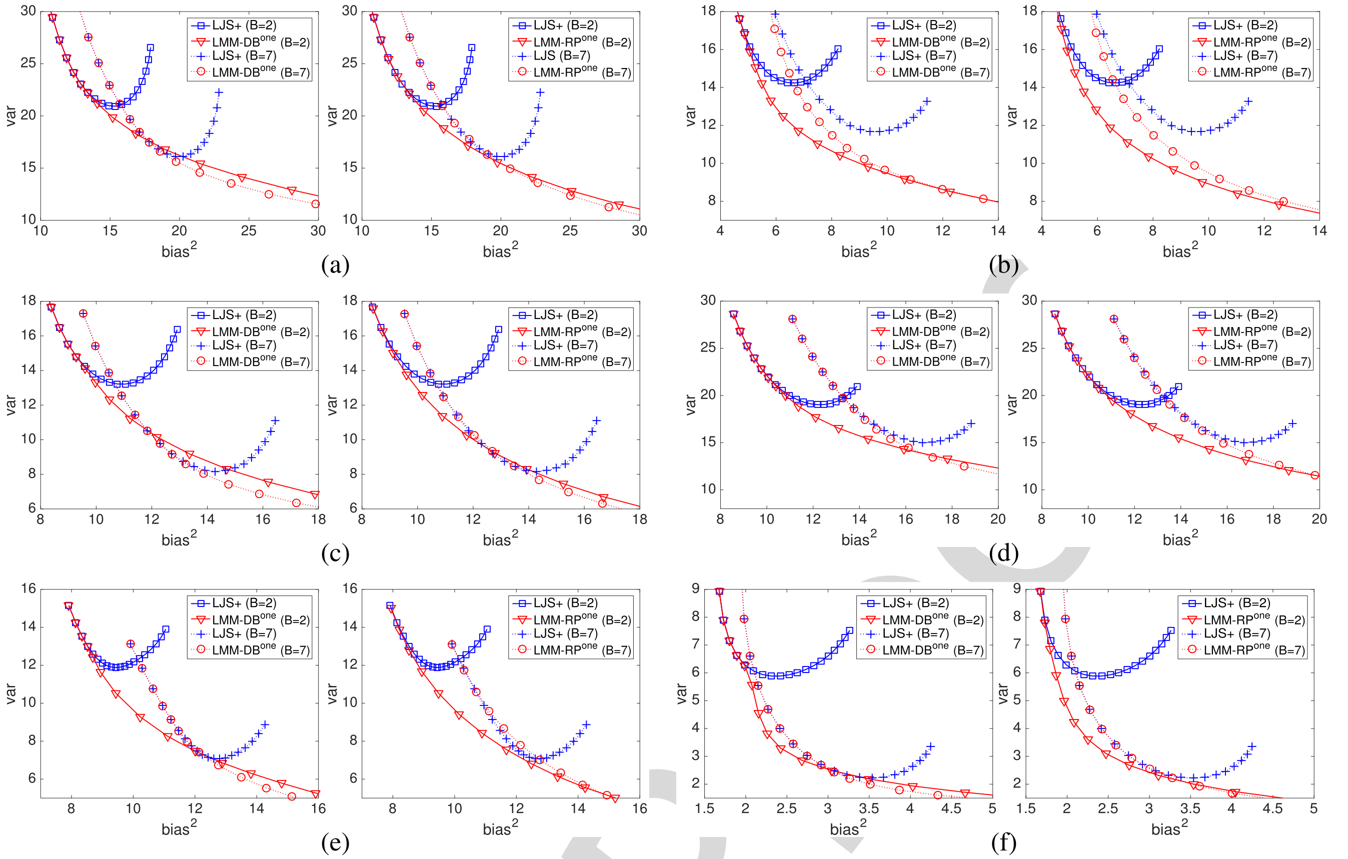



Fig. 4. Bias-variance curves for natural images using $\mathrm{LJS}_+$ [33] and our proposed $\mathrm{LMM-DB}^{\mathrm{one}}$ and $\mathrm{LMM-RP}^{\mathrm{one}}$ methods with a noise level of $\sigma = 10$. (a) *couple*. (b) *montage*. (c) *lena*. (d) *pepper*. (e) *house*. (f) *MRI*.

neighborhood sizes $B$ in order for the LJS method to obtain a certain level of bias, but our proposed methods were able to achieve that same level of bias by adjusting the smoothing parameter $h$, which was the same as in classical NLM. Based on our results, it appears that the use of LMM-RP has slightly more advantages than using LMM-DB in terms of the PSNR, as shown in Table I, and the bias-variance trade-off curves, as shown in Fig. 5, for high noise levels.

### *D. Performance Studies With Visual Quality Assessment*

The most important improvements in our proposed LMM-DB and LMM-RP methods when compared to the LJS method were achieved in terms of the visual quality. Figure 6 (a) shows the true cameraman image (left) and the noisy image (right) with a noise level of $\sigma = 10$. Figure 6 (b) presents the filtered images using the LJS method [33] with $B = 2$ and $B = 7$. Severe artifacts were observed in the background areas when using $B = 2$, and these artifacts were reduced when using $B = 7$. However, there were still some artifacts near the edges of objects. Our proposed LMM-DB and LMM-RP methods exhibited fewer image artifacts than were observed in the images processed using the LJS method for both $B = 2, 7$. This tendency was observed in many of the natural images, as shown in Fig. 7, especially in the high intensity flat areas. PSNR improvements in the LJS method were achieved with severe (when $B = 2$) or mild (when $B = 7$) artifacts; however, our proposed methods achieved both a high PSNR and significantly reduced visual artifacts. This ability to reduce the number of visual artifacts in a denoised image is important in some applications, such as diagnostic medical imaging.

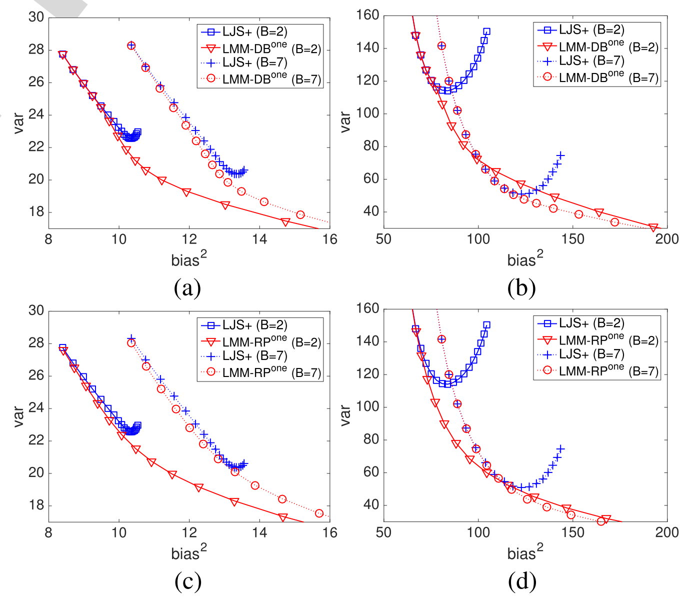



Fig. 5. Bias-variance curves for $\mathrm{LMM-DB}^{\mathrm{one}}$ and $\mathrm{LMM-RP}^{\mathrm{one}}$ for comparison with $\mathrm{LJS}_+$ for two neighborhood sizes $B = 2, 7$ and two noise levels $\sigma = 10, 40$. (a) $\mathrm{LMM-DB}^{\mathrm{one}}(\sigma = 10)$. (b) $\mathrm{LMM-DB}^{\mathrm{one}}(\sigma = 40)$. (c) $\mathrm{LMM-RP}^{\mathrm{one}}(\sigma = 10)$. (d) $\mathrm{LMM-RP}^{\mathrm{one}}(\sigma = 40)$.

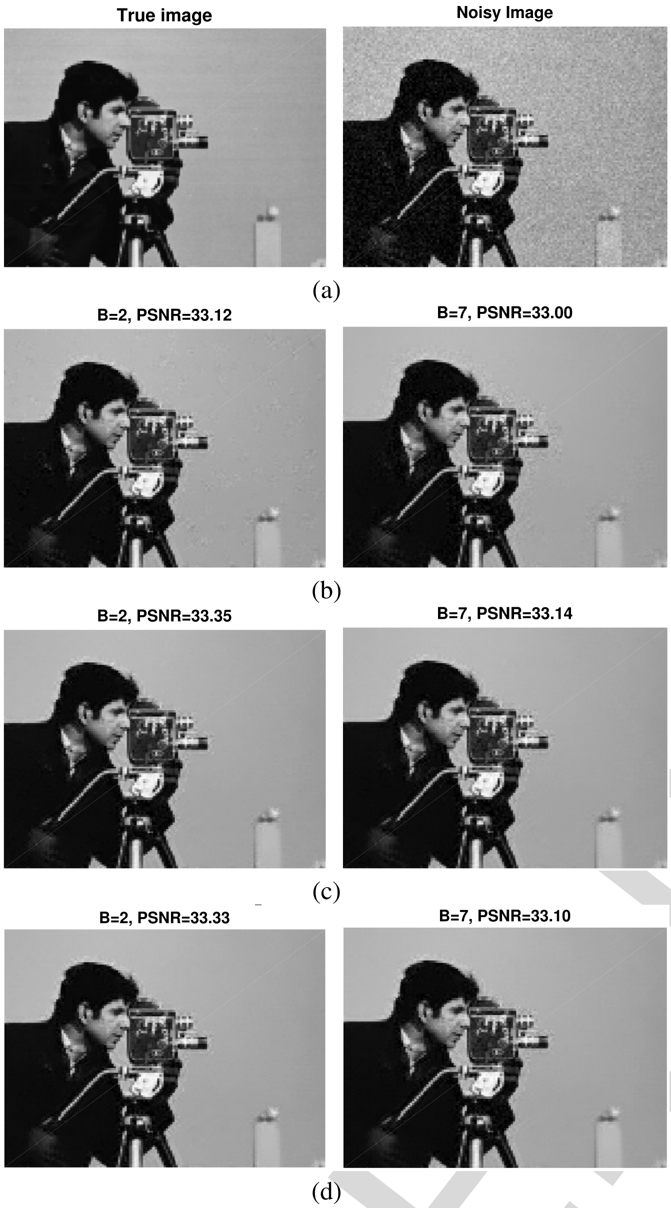



Fig. 6. True, noisy ($\sigma = 10$), and filtered images using $\mathrm{LJS}_+$ [33], and the proposed $\mathrm{LMM-DB^{one}}$ and $\mathrm{LMM-RP^{one}}$. (a) True and noisy images ($\sigma = 10$). (b) $\mathrm{LJS}_+$ [33]. (c) Proposed $\mathrm{LMM-DB^{one}}$. (d) Proposed $\mathrm{LMM-RP^{one}}$.

### E. Maximum Self-Weights: One vs. Stein's

Two maximum self-weights were proposed for use: the value one in (14) that was proposed in [4], and Stein's in (15) that was proposed in [32]. Figure 8 shows that the $\mathrm{LMM-DB^{one}}$ method yielded an improved bias-variance curve and PSNR than did the $\mathrm{LMM-DB^{stein}}$ method when the noise levels were low. For high noise levels $\sigma = 40$, the $\mathrm{LMM-DB^{stein}}$ method yielded an improved PSNR and bias-variance curve than did the $\mathrm{LMM-DB^{one}}$ method. However, these differences were not significant, as also illustrated in terms of the PSNR in Table I. In terms of the visual quality, no significant differences were observed between the two methods.

TABLE II

PERCENTAGE (%) OF $c(\|\mathbf{s}\|)$ THAT EXCEED 2 USING LMM − DB AND LMM − RP METHODS, $\sigma = 10$, B = 2

| | | $\sigma = 10$ | $\sigma = 20$ | $\sigma = 40$ | $\sigma = 60$ |
|---|---|---|---|---|---|
| *cameraman* | $\mathrm{LMM\text{-}DB^{one}}$ | 0.32 | 0.04 | 0.03 | 0.05 |
| | $\mathrm{LMM\text{-}DB^{stein}}$ | 0.85 | 0.66 | 0.21 | 0.18 |
| *fingerprint* | $\mathrm{LMM\text{-}DB^{one}}$ | 0.00 | 0.00 | 0.00 | 0.00 |
| | $\mathrm{LMM\text{-}DB^{stein}}$ | 0.30 | 0.13 | 0.09 | 0.02 |
| *MRI* | $\mathrm{LMM\text{-}DB^{one}}$ | 0.10 | 0.05 | 0.10 | 0.13 |
| | $\mathrm{LMM\text{-}DB^{stein}}$ | 0.18 | 0.16 | 0.16 | 0.16 |

| | | $\sigma = 10$ | $\sigma = 20$ | $\sigma = 40$ | $\sigma = 60$ |
|---|---|---|---|---|---|
| *cameraman* | $\mathrm{LMM\text{-}RP^{one}}$ | 0.25 | 0.04 | 0.01 | 0.00 |
| | $\mathrm{LMM\text{-}RP^{stein}}$ | 1.07 | 0.90 | 0.20 | 0.22 |
| *fingerprint* | $\mathrm{LMM\text{-}RP^{one}}$ | 0.01 | 0.00 | 0.00 | 0.00 |
| | $\mathrm{LMM\text{-}RP^{stein}}$ | 0.27 | 0.19 | 0.13 | 0.03 |
| *MRI* | $\mathrm{LMM\text{-}RP^{one}}$ | 0.09 | 0.05 | 0.07 | 0.09 |
| | $\mathrm{LMM\text{-}RP^{stein}}$ | 0.23 | 0.20 | 0.16 | 0.13 |

### F. "Practical" Minimax Estimator

The proposed LMM-DB and LMM-RP methods are minimax with respect to $\|\mathbf{s}\| \leq \Upsilon_4$ and $\|\mathbf{s}\| \leq \Upsilon_3$, respectively, as shown in Fig. 3. However, these conditions impose upper bounds for the smoothing parameters $h$ and the optimal $h^*$, which means that the smoothing parameter values that yield the best PSNR may not be achievable. We empirically investigated this issue using many natural images.

Table II shows the ratio (percentage unit) of the number of pixels for which $c(\|\mathbf{s}\|) > 2$ to the total number of pixels in the cameraman, fingerprint, and MRI images when the optimal $h^*$ for the highest PSNR was chosen based on the true images for the proposed LMM-DB and LMM-RP methods. For most of the pixels, the LMM-DB and LMM-RP values were minimax. The relationship between the percentage of pixels with $c(\|\mathbf{s}\|) > 2$ and the root mean squared error (RMSE) is illustrated in Fig. 9 for the cameraman and MRI images. Surprisingly, the optimal global smoothing parameters $h$ for the lowest RMSE point (or the highest PSNR) of the LMM-DB and LMM-RP methods are very close to the smoothing parameters $h$ such that the percentage of $c(\|\mathbf{s}\|) > 2$ is 0.1%. This phenomenon was not only observed in these two images. As shown in Table III, the pixel percentage of $c(\|\mathbf{s}\|) > 2$ that do not require knowledge of the true image can still determine smoothing parameters that are able to yield comparable PSNR values to the best PSNR values obtained by using the optimal smoothing parameters calculated based on knowledge of the true image. This was observed in all of the natural images used in our simulations, with different noise levels, and when $B = 2$ was used. However, the criteria of using the pixel percentage of $c(\|\mathbf{s}\|) > 2$ did not work very well for $B = 7$ in our simulations. These criteria can be potentially used when choosing a global smoothing parameter with our proposed methods as a heuristic approach without knowing the true image.

### G. Computation Time for Algorithms

Table IV reports the computation time of the proposed methods in comparison with the classical NLM and $\mathrm{LJS}_+$.

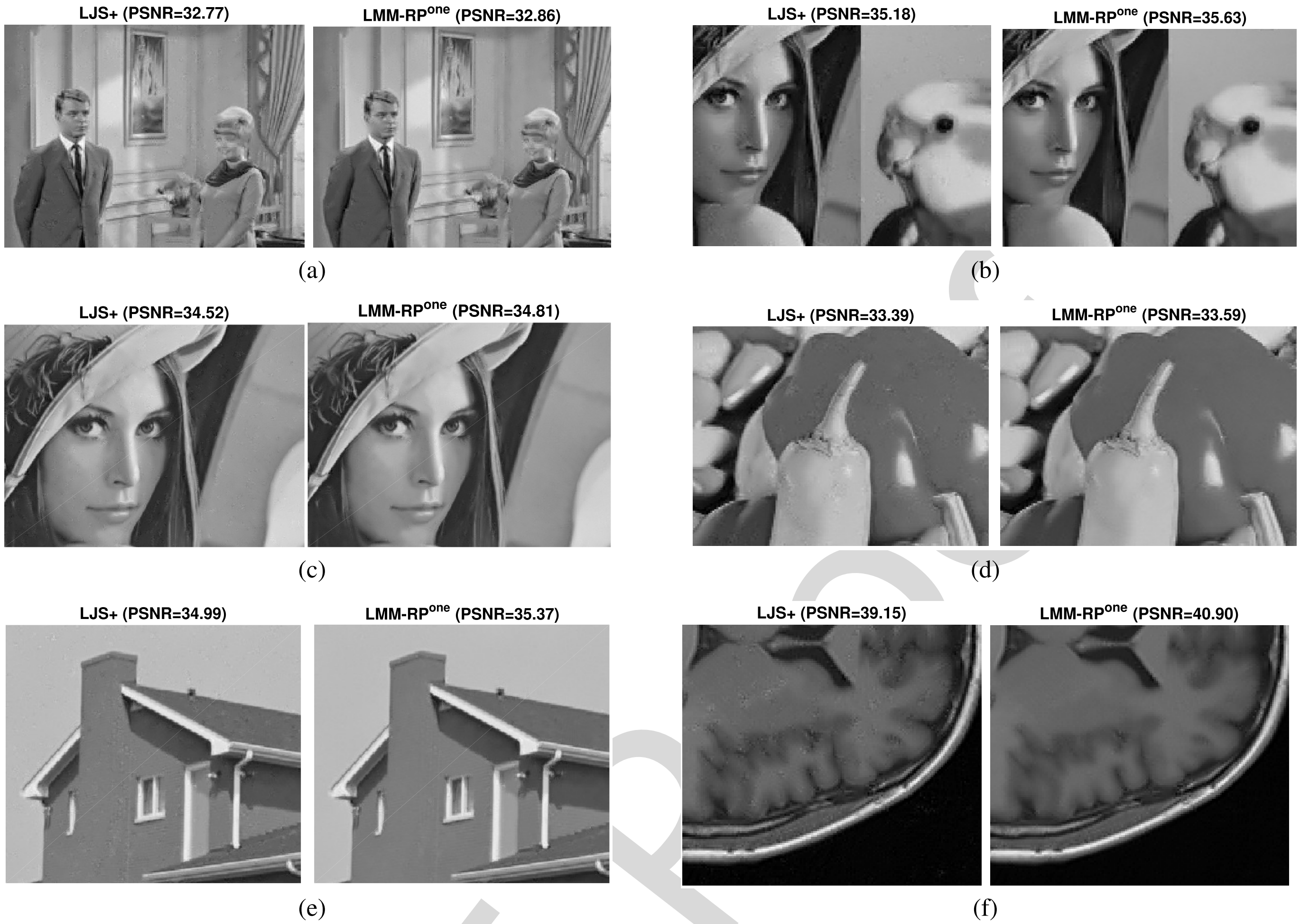



Fig. 7. Filtered results using $\mathrm{LJS}_+$ [33] and the proposed $\mathrm{LJS}-\mathrm{RP}^{\mathrm{one}}$ method with a noise level of $\sigma = 10$ and neighborhood size $\mathrm{B} = 2$. (a) *couple*. (b) *montage*. (c) *lena*. (d) *pepper*. (e) *house*. (f) *MRI*.

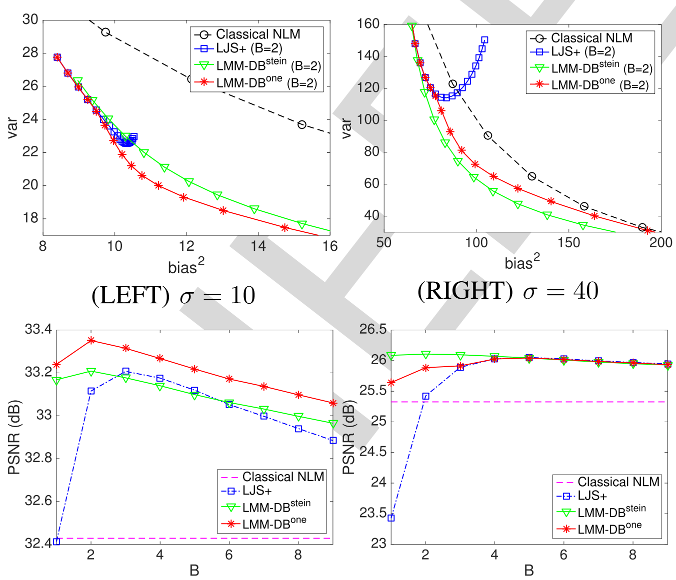



Fig. 8. Bias-variance curves and PSNR vs. varying neighborhood sizes ($B$) using classical NLM (only in the PSNR figure), LJS, and the proposed $\mathrm{LMM}-\mathrm{DB}^{\mathrm{stein}}$ vs. $\mathrm{LMM}-\mathrm{DB}^{\mathrm{one}}$ for the cameraman example.

We used 8 threads (Intel Core i7 2.8 GHz) when computing the patch distances for all methods. The local block size was $B = 2$, the patch size was $7 \times 7$, and the window size was $31 \times 31$. All parameters were fixed for all of results presented in this section. Adjusting these parameters can greatly reduce the running time. For example, setting $B = 4$, the patch size to $5 \times 5$, and the window size to $13 \times 13$ reduces the computation time of the proposed methods to 0.60, 1.12, and 2.91 seconds (s) for $128^2$, $256^2$, and $512^2$ images, respectively. However, analytically, the classical NLM requires $3|\mathbf{P}||\Omega| + 4|\Omega| - 1$ operations per pixel where $|\Omega|$ is the number of elements in $\Omega_i$ and $\mathrm{LJS}_+$ requires $3|\mathbf{P}||\Omega| + 4|\Omega| + 3|\mathbf{B}| + 5$ operations per pixel. It is reported in [33] that the additional operations for $\mathrm{LJS}_+$ ($3|\mathbf{B}| + 6$ operations) were negligible compared to the NLM filtering computation ($3|\mathbf{P}||\Omega| + 4|\Omega| - 1$ operations). Analytically, the additional computation for $\mathrm{LMM}-\mathrm{DB}$ and $\mathrm{LMM}-\mathrm{RP}$ is $3|\mathbf{B}| + 7$ operations, which is almost the same as the additional computation for $\mathrm{LJS}_+$. Therefore, further implementation optimization is possible by exploiting the redundant computation of the patch distances for the minimax estimator and NLM weights.

## VI. Discussion

The classical NLM method was a significant work in image denoising [4], and required the determination of two important parameters for good denoising performance: a smoothing parameter and a self-weight value. The LJS method proposed

TABLE III

THE PSNR VALUES (dB) OF THE PROPOSED METHODS WITH B = 2 WHEN CHOOSING THE SMOOTHING PARAMETER SO AS TO YIELD THE HIGHEST PSNR USING THE TRUE IMAGE (TRUE), AND WHEN CHOOSING THE SMOOTHING PARAMETER SO AS TO YIELD THE PERCENTAGE OF $c(\|\mathbf{s}\|) > 2$ TO BE 0.1% (ESTIMATED) FOR DIFFERENT NOISE LEVELS

| | | LMM-DB $^{\text{one}}$ | | LMM-RP $^{\text{one}}$ | |
|---|---|---|---|---|---|
| | σ | TRUE | ESTIMATED | TRUE | ESTIMATED |
| *cameraman* | 10 | 33.35 | 33.32 | 33.22 | 33.30 |
| | 20 | 29.47 | 29.47 | 29.30 | 29.45 |
| | 40 | 25.91 | 25.90 | 26.16 | 26.01 |
| | 60 | 23.43 | 23.43 | 23.78 | 23.60 |
| *lena* | 10 | 34.72 | 34.74 | 34.73 | 34.78 |
| | 20 | 31.22 | 31.20 | 31.47 | 31.37 |
| | 40 | 27.74 | 27.74 | 28.07 | 27.88 |
| | 60 | 25.60 | 25.63 | 25.95 | 25.82 |
| *montage* | 10 | 35.55 | 35.56 | 35.51 | 35.34 |
| | 20 | 31.24 | 31.20 | 31.33 | 31.32 |
| | 40 | 26.99 | 26.98 | 27.26 | 27.13 |
| | 60 | 24.07 | 23.97 | 24.54 | 24.16 |
| *house* | 10 | 35.32 | 35.35 | 35.30 | 35.37 |
| | 20 | 32.00 | 31.97 | 32.30 | 32.20 |
| | 40 | 27.82 | 27.79 | 28.35 | 28.06 |
| | 60 | 25.23 | 25.23 | 25.75 | 25.37 |
| *peppers* | 10 | 33.56 | 33.54 | 33.48 | 33.59 |
| | 20 | 29.81 | 29.80 | 29.91 | 29.95 |
| | 40 | 25.71 | 25.67 | 26.16 | 25.89 |
| | 60 | 23.15 | 23.02 | 23.81 | 23.24 |
| *barbara* | 10 | 33.67 | 33.68 | 33.62 | 33.73 |
| | 20 | 29.94 | 29.82 | 30.23 | 30.03 |
| | 40 | 25.79 | 25.69 | 26.47 | 25.92 |
| | 60 | 23.56 | 23.48 | 24.14 | 23.67 |
| *boat* | 10 | 32.80 | 32.80 | 32.73 | 32.79 |
| | 20 | 29.22 | 29.18 | 29.36 | 29.22 |
| | 40 | 25.44 | 25.60 | 25.99 | 25.75 |
| | 60 | 23.76 | 23.76 | 24.05 | 23.91 |
| *hill* | 10 | 32.66 | 32.64 | 32.60 | 32.61 |
| | 20 | 29.24 | 29.17 | 29.49 | 29.30 |
| | 40 | 26.01 | 25.99 | 26.38 | 26.05 |
| | 60 | 24.35 | 24.35 | 24.63 | 24.45 |
| *couple* | 10 | 32.82 | 32.80 | 32.79 | 32.81 |
| | 20 | 28.89 | 28.70 | 29.12 | 28.85 |
| | 40 | 25.08 | 25.05 | 25.52 | 25.10 |
| | 60 | 23.26 | 23.19 | 23.56 | 23.28 |
| *fingerprint* | 10 | 30.86 | 30.84 | 30.66 | 30.80 |
| | 20 | 27.07 | 26.86 | 27.05 | 26.96 |
| | 40 | 23.69 | 23.24 | 23.96 | 23.38 |
| | 60 | 21.47 | 20.77 | 21.92 | 21.02 |
| *MRI* | 10 | 40.83 | 40.83 | 40.77 | 40.90 |
| | 20 | 36.59 | 36.60 | 36.71 | 36.74 |
| | 40 | 32.36 | 32.36 | 32.58 | 32.56 |
| | 60 | 29.64 | 29.63 | 29.83 | 29.80 |
| *abbey* | 10 | 30.01 | 29.98 | 29.98 | 29.94 |
| | 20 | 25.89 | 25.74 | 25.91 | 25.60 |
| | 40 | 22.96 | 22.80 | 23.03 | 22.81 |
| | 60 | 21.58 | 21.58 | 21.71 | 21.65 |
| *airplane cabin* | 10 | 32.47 | 32.45 | 32.50 | 32.49 |
| | 20 | 28.51 | 28.35 | 28.61 | 28.45 |
| | 40 | 24.74 | 24.68 | 25.00 | 24.81 |
| | 60 | 23.13 | 23.04 | 23.34 | 23.30 |
| *airport terminal* | 10 | 33.64 | 33.64 | 33.68 | 33.68 |
| | 20 | 29.63 | 29.53 | 29.83 | 29.63 |
| | 40 | 25.19 | 25.04 | 25.51 | 25.07 |
| | 60 | 22.95 | 22.94 | 23.23 | 23.08 |
| *alley* | 10 | 37.94 | 37.87 | 38.28 | 38.06 |
| | 20 | 34.15 | 34.10 | 34.44 | 34.25 |
| | 40 | 31.06 | 31.03 | 31.19 | 31.19 |
| | 60 | 29.53 | 29.19 | 29.59 | 29.49 |
| *amphitheater* | 10 | 33.00 | 32.97 | 33.00 | 32.94 |
| | 20 | 29.18 | 29.08 | 29.31 | 29.13 |
| | 40 | 25.90 | 25.90 | 25.98 | 25.95 |
| | 60 | 24.54 | 24.51 | 24.59 | 24.59 |

by Wu *et al.* [33] developed a state-of-the-art method for self-weight determination using JS estimation [37] and yielded superior results in terms of the PSNR compared to the other existing methods. However, since the LJS method did not impose an upper bound for self-weight estimation, the bias could no longer be controlled by the smoothing parameter, which resulted in visual quality degradation. Our proposed methods based on the Baranchik's minimax theorem [34] yielded comparable PSNR results to the state-of-the-art LJS method. By imposing upper bounds for the self-weights,

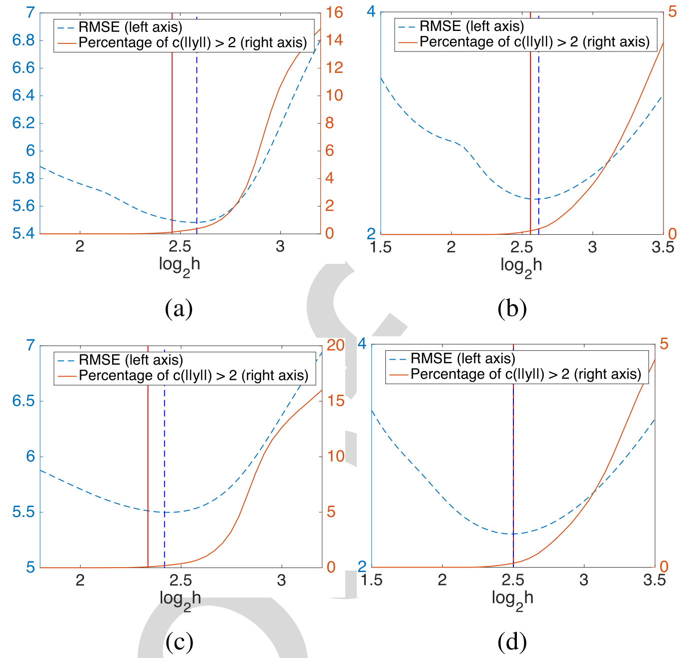



Fig. 9. Comparison plots of the RMSE vs. the smoothing parameter $h$ and the percentage of $c(\|s\|) > 2$ vs. the same smoothing parameter when using LMM-DB and LMM-RP with B = 2 and $\sigma = 10$. (a) cameraman $\mathrm{LMM-DB}^{\mathrm{one}}$. (b) MRI $\mathrm{LMM-DB}^{\mathrm{one}}$. (c) cameraman $\mathrm{LMM-RP}^{\mathrm{one}}$. (d) MRI $\mathrm{LMM-RP}^{\mathrm{one}}$.

TABLE IV

EXECUTION TIME (S) COMPARISON. THIS WILL VARY WITH PARAMETER SELECTION

| Image size | Classical NLM | LJS$_+$ | LMM-DB $^{\text{one}}$ | LMM-RP $^{\text{one}}$ |
|---|---|---|---|---|
| 128*128 | 0.65 | 0.89 | 0.90 | 0.90 |
| 256*256 | 1.57 | 2.37 | 2.39 | 2.38 |
| 512*512 | 4.90 | 7.14 | 7.19 | 7.18 |

the bias-variance trade-off was able to be controlled by a smoothing parameter, and substantial visual artifact reduction was achieved.

The focus of this article was self-weight parameter selection in the classical NLM filter with theoretical justification. As discussed in the Introduction, there are other factors that affect the performance of NLM based filters, and we expect that our proposed methods would not be able to achieve state-of-the-art denoising performance if there were no other optimizations performed except the self-weights. Indeed, our proposed methods with one patch size (non-adaptive neighborhood) and one global smoothing parameter were not able to achieve the level of denoising performance of the state-of-the-art denoising methods such as BM3D [6]. However, when our proposed methods have incorporated some of the other factors into the NLM filters, such as local smoothing parameters and adaptive neighborhoods [10], they have great potential to achieve significantly improved denoising performance.

The minimax property of our proposed methods depends on the choice of smoothing parameters. When using sufficiently small smoothing parameters, the LMM-DB and LMM-RP methods are "practically" minimax according to Baranchik's theorem [34]. However, when large smoothing parameters are used, there may be some pixels that are not minimax for self-weight estimation. More empirical investigation showed that the optimal global smoothing parameter $h$ that yielded the best PSNR only resulted in a very small portion of the pixels that did not have minimax self-weight estimators. In fact, this can be used as a useful heuristic when choosing a good smoothing parameter since testing the minimax properties of our proposed methods does not require the true image. More theoretical analysis for this observation, or a statistical analysis using many natural images as shown in [41], are potential extensions of this work. Therefore, our proposed methods do not only provide an optimal way to determine self-weights, but also provide a heuristic way to determine a good smoothing parameter.

## VII. Conclusion

We proposed two methods, LMM-DB, LMM-RP, to determine the self-weights of NLM filters that are "practically" minimax, and this methods yielded a comparable PSNR, better bias-variance trade-offs, and reduced visual quality artifacts when compared to the results obtained using the state-of-the-art LJS method. Our methods also provide a potentially useful heuristic way to determine a global smoothing parameter without knowledge of the original image.

## References


[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Mumbai, India, Jan. 1998, pp. 839–846.

[2] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.

[3] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

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

[5] S. P. Awate and R. T. Whitaker, "Unsupervised, information-theoretic, adaptive image filtering for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 364–376, Mar. 2006.

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

[7] C. Louchet and L. Moisan, "Total variation as a local filter," *SIAM J. Imag. Sci.*, vol. 4, no. 2, pp. 651–694, Jan. 2011.

[8] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2012.

[9] D. Barash, "Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 844–847, Jun. 2002.

[10] 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.

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

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

[13] 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.

[14] Z. Ji, Q. Chen, Q.-S. Sun, and D.-S. Xia, "A moment-based nonlocal-means algorithm for image denoising," *Inf. Process. Lett.*, vol. 109, nos. 23–24, pp. 1238–1244, Nov. 2009.

[15] S. Grewenig, S. Zimmer, and J. Weickert, "Rotationally invariant similarity measures for nonlocal image denoising," *J. Vis. Commun. Image Represent.*, vol. 22, no. 2, pp. 117–130, Feb. 2011.

[16] R. Yan, L. Shao, S. D. Cvetković, and J. Klijn, "Improved nonlocal means based on pre-classification and invariant block matching," *J. Display Technol.*, vol. 8, no. 4, pp. 212–218, Apr. 2012.

[17] C. A. Deledalle, V. Duval, and J. Salmon, "Non-local methods with shape-adaptive patches (NLM-SAP)," *J. Math. Imag. Vis.*, vol. 43, no. 2, pp. 103–120, Jun. 2012.

[18] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1635–1649, Apr. 2012.

[19] S. Y. Chun, J. A. Fessler, and Y. K. Dewaraja, "Post-reconstruction nonlocal means filtering methods using CT side information for quantitative SPECT," *Phys. Med. Biol.*, vol. 58, no. 17, pp. 6225–6240, 2013.

[20] C. Chan, R. Fulton, R. Barnett, D. Feng, and S. Meikle, "Postreconstruction nonlocal means filtering of whole-body PET with an anatomical prior," *IEEE Trans. Med. Imag.*, vol. 33, no. 3, pp. 636–650, Mar. 2014.

[21] 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.

[22] D. Van De Ville and M. Kocher, "Nonlocal means with dimensionality reduction and SURE-based parameter selection," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2683–2690, Sep. 2011.

[23] C.-A. Deledalle, F. Tupin, and L. Denis, "Poisson NL means: Unsupervised non local means for Poisson noise," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 801–804.

[24] Y. Wu, B. H. Tracey, P. Natarajan, and J. P. Noonan, "Fast blockwise SURE shrinkage for image denoising," *Signal Process.*, vol. 103, pp. 45–59, Oct. 2014.

[25] B. Goossens, Q. Luong, A. Pižurica, and W. Philips, "An improved non-local denoising algorithm," in *Proc. Int. Workshop Local Non-Local Approx. Image Process.*, Lausanne, Switzerland, 2008, pp. 143–156.

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

[27] C. Kervrann, J. Boulanger, and P. Coupé, "Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal," in *Scale Space and Variational Methods in Computer Vision*. Berlin, Germany: Springer, 2007, pp. 520–532.

[28] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, "Probabilistic nonlocal means," *IEEE Signal Process. Lett.*, vol. 20, no. 8, pp. 763–766, Aug. 2013.

[29] V. Dore 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.

[30] T. Brox and D. Cremers, "Iterated nonlocal means for texture restoration," in *Scale Space and Variational Methods in Computer Vision*. Berlin, Germany: Springer, 2007, pp. 13–24.

[31] 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 Approx. Image Process.*, 2008, pp. 135–142.

[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] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, "James–Stein type center pixel weights for non-local means image denoising," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 411–414, Apr. 2013.

[34] A. J. Baranchik, "A family of minimax estimators of the mean of a multivariate normal distribution," *Ann. Math. Statist.*, vol. 41, no. 2, pp. 642–645, 1970.

[35] M. P. Nguyen and S. Y. Chun, "Center pixel weight estimation for non-local means filtering using local James-Stein estimator with bounded self-weights," in *Proc. IEEE 13th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2016, pp. 82–85.

[36] C. Stein, "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution," in *Proc. 3rd Berkeley Symp. Math. Statist. Probab.*, vol. 1. 1956, pp. 197–206.

[37] W. James and C. Stein, "Estimation with quadratic loss," in *Proc. 4th Berkeley Symp. Math. Statist. Probab.*, vol. 1. 1961, pp. 361–379.

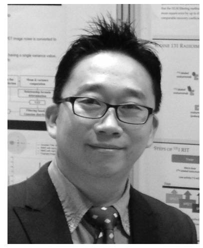
[38] A. J. Baranchik, "Multiple regression and estimation of the mean of a multivariate normal distribution," Dept. Statist., Stanford Univ., Stanford, CA, USA, Tech. Rep. 51, May 1964.

[39] B. Efron and C. Morris, "Families of minimax estimators of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 4, no. 1, pp. 11–21, 1976.

[40] W. E. Strawderman, "Proper Bayes minimax estimators of the multivariate normal mean vector for the case of common unknown variances," *Ann. Statist.*, vol. 1, no. 6, pp. 1189–1194, 1973.

[41] X. Li, H. He, R. Wang, and D. Tao, "Single image superresolution via directional group sparsity and directional features," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2874–2888, Sep. 2015.

**Minh Phuong Nguyen** received the B.S.E. degree in biomedical electronics from the School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam, in 2013, and the M.S. degree in electrical engineering from the Ulsan National Institute of Science and Technology, Ulsan, South Korea. He is currently pursuing the Ph.D. degree with the Delft University of Technology, The Netherlands. His research interests include image reconstruction and image filtering for SPECT, image registration between histology images and MRI, and electrocardiogram signal processing for authentication tasks.



**Se Young Chun** (M'10) received the B.S.E. degree from the School of Electrical Engineering, Seoul National University, Seoul, South Korea, in 1999, and the M.S.E. degree in electrical engineering systems, the M.S. degree in mathematics, and the Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, MI, USA, in 2005 and 2009, respectively.

From 2003 to 2005, he was a recipient of the two year scholarship from the Ministry of Information and Communications, South Korea. From 2009 to 2011, he was a Research Fellow with Massachusetts General Hospital and Harvard Medical School, Boston, MA, USA, and from 2011 to 2013, he was a Research Fellow with Electrical Engineering and Computer Science and Radiology, University of Michigan. Since 2013, he has been an Assistant Professor with the School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan, South Korea. His research interests include image reconstruction, image denoising, image registration for multi-modal imaging systems, motion compensated image reconstruction, and statistical bio signal processing and biometrics.

Dr. Chun was a recipient of the 2010 Society of Nuclear Medicine Computer and Instrumentation Young Investigator Award, 2nd place. He was also a recipient of the 2015 Bruce Hasegawa Young Investigator Medical Imaging Science Award from the IEEE Nuclear and Plasma Sciences Society. He is a member of IEEE Computational Imaging Special Interest Group and an Editorial Board Member of *Nuclear Medicine and Molecular Imaging* (Springer).

# Bounded Self-Weights Estimation Method for Non-Local Means Image Denoising Using Minimax Estimators

Minh Phuong Nguyen and Se Young Chun, *Member, IEEE*

***Abstract*— A non-local means (NLM) filter is a weighted average of a large number of non-local pixels with various image intensity values. The NLM filters have been shown to have powerful denoising performance, excellent detail preservation by averaging many noisy pixels, and using appropriate values for the weights, respectively. The NLM weights between two different pixels are determined based on the similarities between two patches that surround these pixels and a smoothing parameter. Another important factor that influences the denoising performance is the self-weight values for the same pixel. The recently introduced local James-Stein type center pixel weight estimation method (LJS) outperforms other existing methods when determining the contribution of the center pixels in the NLM filter. However, the LJS method may result in excessively large self-weight estimates since no upper bound is assumed, and the method uses a relatively large local area for estimating the self-weights, which may lead to a strong bias. In this paper, we investigated these issues in the LJS method, and then propose a novel local self-weight estimation methods using direct bounds (LMM-DB) and reparametrization (LMM-RP) based on the Baranchik's minimax estimator. Both the LMM-DB and LMM-RP methods were evaluated using a wide range of natural images and a clinical MRI image together with the various levels of additive Gaussian noise. Our proposed parameter selection methods yielded an improved bias-variance trade-off, a higher peak signal-to-noise (PSNR) ratio, and fewer visual artifacts when compared with the results of the classical NLM and LJS methods. Our proposed methods also provide a heuristic way to select a suitable global smoothing parameters that can yield PSNR values that are close to the optimal values.**

***Index Terms*— James-Stein estimator, minimax estimator, non-local means, center pixel weight, bounded self-weight, image denoising.**

## I. Introduction

IMAGE denoising is a fundamental task in image processing, low-level computer vision, and medical imaging algorithms. The goal of denoising is to suppress image noise when restoring desired details using prior information about the images. For example, based on prior information regarding "smooth images", a simple filter, such as a Gaussian filter, can be designed as a weighted average of the image intensities of the pixels in the local neighborhood with non-adaptive weights. However, this type of filter blurs the edges and details of images because these features are not captured in the assumed prior information. Many edge-preserving denoising methods have been proposed, including bilaterial filters [1], [2], anisotropic diffusion [3], non-local means (NLM) filters [4], [5], collaborative filters (BM3D) [6], and total variation filters [7]. Many filters, including bilaterial filters, anisotropic diffusion, and NLM filters (but, not BM3D, see [8]), can be represented as the weighted averages of adaptive weights or adaptive smoothing [9]. It should be noted that it is important to select appropriate weights in these types of filters in order to obtain improved denoised image quality [8].

Classical NLM filters use the similarities between two local patches in a noisy image to determine the weights in non-local adaptive smoothing [4]. The NLM weights are obtained by first calculating the Euclidean distance between the two local patches, which is denoted $d$, and then by evaluating $\exp(-d^2/h^2)$, where $h$ is a smoothing parameter. This method allows higher weights to be assigned to pixels with similar patches so that edges and details can be preserved through non-local weighted averaging.

There are four different factors that determine the output image quality of a NLM filter in terms of weights. 1) The first factor is the similarity measure $d$. The Euclidean distance is a usual choice [4], but other similarity measures have also been proposed, such as hypothesis testing with adaptive neighborhoods [10], principal component analysis (or the subspace based method) [11], [12], blockwise aggregation [13], rotation-invariant measures [14]–[16], shape-adaptive patches [17], and patch-based similarities with adaptive neighborhoods [18]. In multimodal medical imaging, inaccurate weights for noisy molecular images were enhanced by using additional high quality anatomical images [19], [20]. 2) The second factor is the strategy for determining the smoothing parameter $h$. Optimization strategies have been developed based on Stein's unbiased risk estimation (SURE) method for NLM with Gaussian noise [21], [22], NLM with Poisson noise [23], and blockwise NLM with Gaussian noise [24]. 3) The third factor is in selecting the function to use to determine the weights,

Manuscript received March 9, 2016; revised October 26, 2016 and December 31, 2016; accepted January 21, 2017. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning under Grant NRF-2014R1A1A1007928. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Oleg V. Michailovich. *(Corresponding author: Se Young Chun)*

Minh Phuong Nguyen was with the School of Electrical and Computer Engineering (ECE), Ulsan National Institute of Science and Technology (UNIST), Ulsan, 44919, South Korea. Now he is with Section of Radiation, Detection and Medical Imaging, Delft University of Technology, Mekelweg 15, 2629 JB Delft, The Netherlands (e-mail: m.p.nguyen@tudelft.nl).

Se Young Chun is with the School of ECE, UNIST, Ulsan, 44919, South Korea (e-mail: sychun@unist.ac.kr).

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.2017.2658941

such as $\exp(-x^2)$. Other functions have also been proposed to calculate the weights, such as compact support functions [25], [26] and statistical distance functions [27], [28]. 4) The last factor, which is the focus of this article, is in how best to determine the self-weights for the same pixel in the input and output images.

The NLM weights for two different pixels are essentially determined by the distance between the two noisy local patches around these pixels. However, the weights for the same pixel, or the self-weights, are not affected by the noise in the patches and the distance is always 0. For an extremely noisy image, the self-weights will be relatively too large when compared to the other weights, which will cause the filter output to be almost the same as the input noisy image. Therefore, the use of appropriate self-weight values can significantly affect the quality of the denoised image. Many researchers have investigated strategies for determining the self-weights, which are also known as center pixel weights, in order to alleviate the so-called “rare patch effect.” For the classical NLM filter proposed by Buades *et al.*, the self-weights were set to be either one or the maximum weight in a neighborhood [4]. This strategy guaranteed that at least one or two of the largest weights would be the same. Doré and Cheriet also used the maximum weight in a neighborhood as the self-weight, but only if that maximum weight was large [29]. Brox and Cremers proposed a method to have at least $n$ number of the weights to be the same [30], and Zimmer *et al.* considered the self-weight to be a free parameter during the estimation process [31]. Salmon developed a SURE-based method for determining the self-weights that accounted for the noise [32].

Recently, Wu *et al.* proposed a method to determine the self-weights using a James-Stein (JS) type estimator [33]. The idea of that work was to use a JS estimator to determine the reparametrized self-weight in a local neighborhood (called the local JS estimator (LJS)). The LJS method yielded the best peak signal-to-noise ratio (PSNR) results when compared to other existing self-weight selection methods [4], [32]. However, the method had some limitations. First, the LJS could yield self-weights that were theoretically much larger than 1 because no upper bound for the self-weights was assumed, and this may lead to severe rare patch artifacts. The JS estimator does not guarantee its optimality for bounded shrinkage parameters. Second, the original LJS method was tested with a relatively large local neighborhood when determining a self-weight because it was assumed that the self-weights were the same in the local neighborhood. However, the problem is that the selection of a local neighborhood size that is too large may introduce a strong bias into the resulting denoised images.

In this article, we investigate the original LJS method in terms of the local neighborhood size for self-weight estimation and the potential for excessive self-weight estimation when no upper bound is applied on the self-weight. We then propose novel self-weight estimation methods for NLM that account for bounded self-weights using Baranchik’s minimax estimator [34], called local minimax self-weight estimation with direct bound (LMM-DB) and with reparametrization (LMM-RP). We evaluated our proposed methods using performance criteria including PSNR, the bias-variance trade-off curve and visual quality assessment with a wide range of natural images and a real patient MRI image with various noise levels. We compared the performance of our proposed methods with a classical NLM filter using self-weights of 1 [4] and the state-of-the-art LJS method, which has already been shown to be the best among all other previous self-weight determination methods [33].

This article is an extension of a work that was presented at the 2016 IEEE International Symposium on Biomedical Imaging (ISBI) [35], and goes into more depth regarding the theory of the minimax estimator and provides an evaluation of the methods using a significantly larger image dataset.

This paper is organized as follows. Section II reviews the classical NLM filter and revisits the LJS method. Section III investigates the LJS method in terms of the local neighborhood size for self-weight estimation and the potential for excessively large self-weight estimates. Then, Section IV proposes novel LMM-DB and LMM-RP methods using Baranchik’s minimax estimator in order to overcome two limitations of the LJS method. Section V illustrates the performance of our proposed methods by providing our simulation results. Lastly, Sections VI and VII discuss and then conclude this paper, respectively.

## II. Review of the Local James-Stein Self-Weight Estimation Method for the NLM Filter

In this section, we will briefly review both the classical NLM method proposed by Buades *et al.* [4] and the LJS self-weight selection method proposed by Wu *et al.* [33].

### A. Reviewing the Classical Non-Local Means Filter

Let us assume that an image $\mathbf{x}$ is contaminated by noise $\mathbf{n}$, which produces a noisy image $\mathbf{y}$:

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

where $\mathbf{n}$ is zero-mean white Gaussian noise with standard deviation $\sigma$. The NLM filtered value at the pixel $i$ is the weighted average of all pixels in a search region $\boldsymbol{\Omega}_i$:

$$\hat{x}_i = \frac{\sum_{j\in\Omega_i} w_{i,j}\, y_j}{\sum_{j\in\Omega_i} w_{i,j}} \tag{2}$$

where $y_i$ is the $i$th element of $\mathbf{y}$, $w_{i,j}$ is the weight between the $i$th and $j$th pixels, and $\boldsymbol{\Omega}_i$ is the set of all pixels in an area around the $i$th pixel, which could be an entire image. The similarity weight of the classical NLM is defined as:

$$w_{i,j} = \exp\left(\frac{-\left\|\mathbf{P}_i\mathbf{y} - \mathbf{P}_j\mathbf{y}\right\|^2}{2\,|\mathbf{P}|\,h^2}\right) \tag{3}$$

where $\mathbf{P}_i$ is an operator used to extract a square-shaped patch centered at the $i$th pixel, $\|\cdot\|$ is an $l_2$ norm, $|\mathbf{P}|$ is the number of pixels within a patch, and $h$ is a global smoothing parameter. Equation (3) implies that the self-weights $w_{i,i}$ are always equal to 1. Previous works on self-weights have shown that good strategies for determining the self-weights also affect the image quality of the NLM filtering [4], [29], [32], [33].

### B. Reviewing Local James-Stein Self-Weight Estimation

The LJS method was proposed in order to determine $w_{i,i}$ as follows [33]. First, (2) was decomposed into two terms:

$$\hat{x}_i = \frac{W_i}{W_i + w_{i,i}} \hat{z}_i + \frac{w_{i,i}}{W_i + w_{i,i}} y_i \tag{4}$$

where $W_i = \sum_{j \in \Omega_i \setminus \{i\}} w_{i,j}$ and

$$\hat{z}_i = \sum\nolimits_{j \in \Omega_i \setminus \{i\}} w_{i,j}\, y_j / W_i. \tag{5}$$

The terms $\hat{z}_i$ do not contain $w_{i,i}$. Then, the LJS method reparametrized (4) using

$$p_i = \frac{w_{i,i}}{W_i + w_{i,i}} \tag{6}$$

so that (4) became:

$$\hat{x}_i = (1 - p_i)\, \hat{z}_i + p_i\, y_i. \tag{7}$$

The problem of estimating the self-weights $w_{i,i}$ became the problem of estimating $p_i$. Lastly, the JS estimator [36], [37] for $p_i$ was proposed:

$$p_i^{\mathrm{LJS}} = 1 - \frac{(|\mathbf{B}| - 2)\, \sigma^2}{\left\| \mathbf{B}_i \mathbf{y} - \mathbf{B}_i \hat{\mathbf{z}} \right\|^2} \tag{8}$$

where $\mathbf{B}_i$ is an operator used to extract a square-shaped neighborhood centered at the $i$th pixel, $|\mathbf{B}|$ is the number of pixels within that neighborhood, and $\sigma$ is the known noise level.

Equation (8) implies that $p_i^{\mathrm{LJS}} \in (-\infty, 1]$. Since the weights are non-negative, it was proposed to use the zero-lower bound for $p_i^{\mathrm{LJS}}$ as follows [33]:

$$\hat{x}_i^{\mathrm{LJS}_+} = \left(1 - p_i^{\mathrm{LJS}_+}\right) \hat{z}_i + p_i^{\mathrm{LJS}_+} y_i \tag{9}$$

where

$$p_i^{\mathrm{LJS}_+} := [p_i^{\mathrm{LJS}}]_+ = \left[ 1 - \frac{(|\mathbf{B}| - 2)\, \sigma^2}{\left\| \mathbf{B}_i \mathbf{y} - \mathbf{B}_i \hat{\mathbf{z}} \right\|^2} \right]_+ \tag{10}$$

and $[s]_+ := \max\ (s, 0)$. Wu *et al.* also mentioned that a user-defined upper bound for $p_i$ can be used, but did not investigate further [33]. It should be noted that the JS estimator does not guarantee its optimality when bounding $p_i^{\mathrm{LJS}}$ in (8).

## III. Limitations of the Local James-Stein Self-Weight Estimation for the NLM Filter

We now investigate two limitations of the original LJS method [33] in terms of the size of local neighborhoods for self-weight estimation, and the potential for excessive self-weight estimation.

### A. Size of Local Neighborhood for Self-Weight Estimation

In the method described in [33], there are two implicit steps required in order to obtain the LJS self-weight estimator (10). The first step is to choose a local set of pixels around the $i$th pixel, referred to as set $\Omega_i^B$, that correspond to the operator $\mathbf{B}_i$, and assume that:

$$\hat{x}_j = (1 - p_i)\, \hat{z}_j + p_i\, y_j, \quad j \in \Omega_i^B. \tag{11}$$

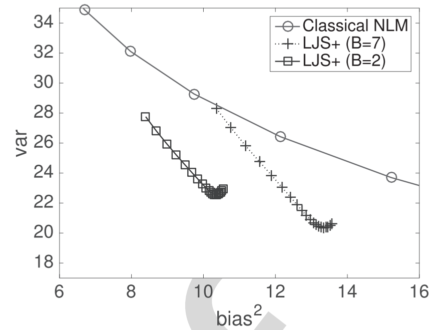



Fig. 1. Bias-variance curves (cameraman example) for the classical NLM and LJS methods ($\mathrm{LJS}_+$) for different sizes of local neighborhoods (B). The curves were plotted while varying the smoothing parameter $h$ ($log_2\ h \in [1.8, 3.2]$).

Based on the works of Stein [36] and James and Stein [37], if $|\mathbf{B}| \geq 3$, then for a neighborhood $\Omega_i^B$ extracted using $\mathbf{B}_i$,

$$\hat{x}_j = \left(1 - p_i^{\mathrm{LJS}_+}\right) \hat{z}_j + p_i^{\mathrm{LJS}_+} y_j, \quad j \in \Omega_i^B. \tag{12}$$

is a dominant estimator for $x_j$ "locally" in $\Omega_i^B$. The LJS method used the zero lower bound when estimating $p_i$ in order to obtain a realistic non-negative self-weight value. This was also a good choice in terms of the estimator performance since the positive part of the JS estimator is dominant over the original JS estimator, according to the works of Baranchik [34], [38] and Efron and Morris [39].

The second implicit step is to assign the resulting $p_i^{\mathrm{LJS}}$ to $p_i$ in (7) for only the single pixel $i$ so that:

$$\hat{x}_i^{\mathrm{LJS}} = \left(1 - p_i^{\mathrm{LJS}}\right) \hat{z}_i + p_i^{\mathrm{LJS}} y_i. \tag{13}$$

Wu *et al.* evaluated the LJS method with $|\mathbf{B}| = 15 \times 15$ [33], which seems relatively large.

Based on this implicit two-step interpretation, we can surmise that using a smaller size of $|\mathbf{B}|$ may be more desirable for obtaining a less biased estimate of $p_i$ since the assumption of having the same $p_i$ in $\Omega_i^B$ is less likely to be true for larger sizes of $\Omega_i^B$. Figure 1 confirms our conjecture. The bias-variance curves of the LJS method yielded better bias-variance trade-offs than those in the classical NLM method for both large local neighborhoods with a half window size $B = 7$ ($|\mathbf{B}| = 15 \times 15$) and small local neighborhoods with $B = 2$ ($|\mathbf{B}| = 5 \times 5$). However, using larger local neighborhood sizes for estimating $p_i$ yielded stronger biases than those estimated using smaller sizes for the same level of variance.

### B. Excessively Large Self-Weight Estimates

In the LJS method for determining the self-weights by estimating values for $p_i$ [33], it is theoretically possible that the self-weights have excessively high values. For example, (6) suggests that if $p_i = 1$ and $W_i > 0$, then $w_{i,i} \gg 1$. Slight artifacts were observed in [33] in the background area that were potentially caused by excessive self-weight estimates when a relatively larger neighborhood size $|\mathbf{B}| = 15 \times 15$ was used. We observed a significantly higher degree of degradation in the visual image quality in the background area when the

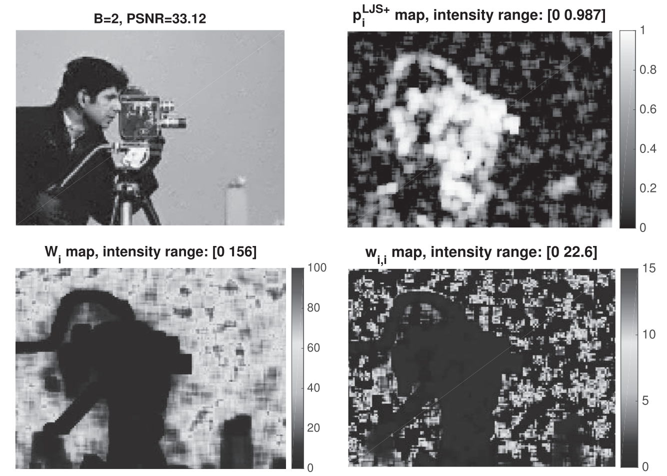



Fig. 2. Denoised image of the cameraman example using the original LJS method [33] with no upper bound for the self-weights (top left), estimated $p_i$ values (top right), calculated $W_i$ (bottom left), and resulting self-weights ($w_{i,i}$) showing excessive self-weights (bottom right). B = 2 and $\sigma = 10$.

size of $|\mathbf{B}|$ in (8) was small, as shown in the image in the top left figure of Fig. 2.

We investigated this issue using an example of the cameraman image that was denoised using the LJS method [33], but with a smaller neighborhood size $|\mathbf{B}| = 5 \times 5$. For areas with more details, such as edges and textures, large $p_i$ values were estimated and yielded large self-weights, as shown in the top right figure of Fig. 2. However, since the values for $W_i$ were also very small in these areas, as shown in the bottom left image in Fig. 2, the resulting self-weight map yielded values close to 1 in the areas with details as shown in the bottom right image in Fig. 2.

In contrast, for areas with almost no details, such as those with a flat intensity background, relatively smaller $p_i$ values were estimated, some of which were much larger than 0 while the rest were closer to 0, as shown in the top right image in Fig. 2. However, since the $W_i$ values for the flat areas were relatively large, as shown in the bottom left image in Fig. 2, some of the estimated $p_i$ values obtained using the LJS method (LJS+) were estimated to yield excessively large self-weights that were much larger than 1, as shown in the bottom right image of Fig. 2. Consequently, these excessively large self-weights caused severe rare patch artifacts in the filtered image, which resulted in visual quality degradation, as observed in the top left image of Fig. 2.

## IV. Local Minimax Estimation Methods for Upper Bounded Self-Weights in a NLM Filter

In this section, we propose two local upper bounded self-weight estimation methods that use Baranchik's minimax estimator [34].

### A. Bounded Self-Weights

It is usually assumed that the self-weights satisfy $w_{i,i} \in [0, 1]$. However, there are many possible upper bounds for the self-weights, including 1 [4] or some positive value that is possibly less than 1 based on SURE [32]. In this article, two different upper bound values $w_{i,i}^{\max}$ for the self-weights were evaluated such that $0 \leq w_{i,i} \leq w_{i,i}^{\max}$. One upper bound was:

$$w_{i,i}^{\max-\mathrm{one}} = 1, \tag{14}$$

which is the usual choice for the self-weights in the classical NLM method [4]. The other upper bound was:

$$w_{i,i}^{\max-\mathrm{stein}} = \exp\left(-\sigma^2/h^2\right), \tag{15}$$

which was motivated by the SURE-based NLM self-weights [32]. We assume that $\sigma$ is known and $h$ is pre-determined, which means that the upper bound for the self-weights can also be determined in advance. Equation (15) takes the noise level into account. As $\sigma$ is smaller, the maximum self-weight in (15) is closer to one. It should be noted that the difference between (15) and (14) will be greater at higher noise levels.

Since $p_i$ is estimated instead of $w_{i,i}$, it is necessary to derive the range of $p_i$ that corresponds to $0 \leq w_{i,i} \leq w_{i,i}^{\max}$. From (6), the derivative of $p_i$ with respect to $w_{i,i}$ is non-negative as follows:

$$\frac{d}{dw_{i,i}} p_i = \frac{W_i}{(W_i + w_{i,i})^2} \geq 0$$

since $W_i \geq 0$. Therefore, $p_i$ is a non-decreasing function of $w_{i,i}$ and for $0 \leq w_{i,i} \leq w_{i,i}^{\max}$, the range of $p_i$ will be

$$0 \leq p_i \leq \frac{w_{i,i}^{\max}}{W_i + w_{i,i}^{\max}} =: p_i^{\max} \leq 1.$$

Note that if $W_i = 0$, then $p_i^{\max} = 1$. The estimator $p_i^{\mathrm{LJS}+}$ in (10) automatically guarantees that $0 \leq p_i \leq 1$ if $|\mathbf{B}| \geq 2$. However, since $W_i > 0$ generally holds for most real images with noise, it is necessary to constrain $p_i$ to be less than or equal to the upper bound $p_i^{\max}$, which is usually less than one.

### B. Local Minimax Self-Weight Estimation With Direct Bound

Enforcing the upper limit $p_i^{\max}$ on the estimated $p_i$ in (10) using $\min(p_i^{\mathrm{LJS}+}, p_i^{\max})$ breaks the optimality of the JS estimator if $p_i^{\max} < 1$. In this article, we propose using Baranchik's minimax estimator [34] to incorporate bounded self-weights into the estimator (see Baranchik [34], Erfon and Morris [39], and Strawderman [40] for more details on this minimax estimator).

*Theorem 1 (Baranchik):* For $\mathbf{y} \sim \mathcal{N}_r\left(\mathbf{x}, \sigma^2\mathbf{I}\right)$, $r \geq 3$, and loss $L(\mathbf{x}, \hat{\mathbf{x}}) = \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|$, an estimator of the form $\hat{\mathbf{x}} = q\mathbf{y}$ where

$$q = \left[1 - c\left(\|\mathbf{y}\|\right) \frac{\sigma^2(r-2)}{\|\mathbf{y}\|^2}\right] \tag{16}$$

is the minimax, provided that:

(i) $0 \leq c\left(\|\mathbf{y}\|\right) \leq 2$ and

(ii) the function $c(\cdot)$ is nondecreasing.

Here $\mathbf{y}$ shrinks toward $\mathbf{0}$ which is the initial estimate of $\mathbf{x}$.

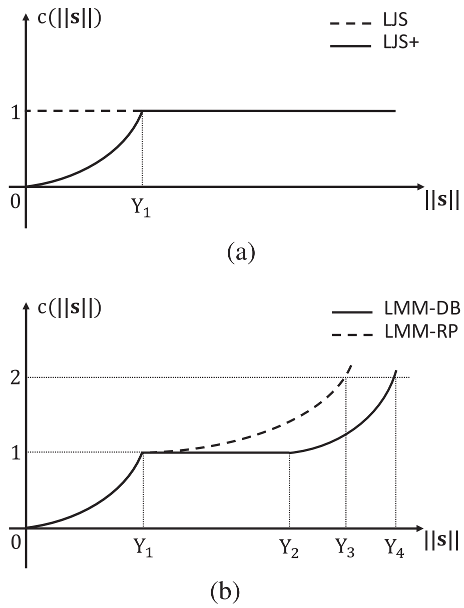



Fig. 3. Graphical illustrations of the original and positive part JS estimators without upper bounds, and the proposed minimax self-weight estimators with upper bounds in terms of $c(\|\mathbf{s}\|)$ vs. $\|\mathbf{s}\|$. (a) Original and positive-part JS estimators. (b) Proposed minimax estimators with bounds.

The original JS estimator and its positive part are special cases of Baranchik's minimax estimator. For the original JS estimator (8):

$$c(\|\mathbf{s}\|) = 1, \tag{17}$$

where $\mathbf{s} = \mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}$ so that both conditions (i, ii) of the Baranchik's theorem are satisfied. In the positive part estimator (9), it can be shown that:

$$c(\|\mathbf{s}\|) = \begin{cases} \dfrac{\|\mathbf{s}\|^2}{\sigma^2(r-2)}, & 0 \le \|\mathbf{s}\| \le \mathrm{Y}_1 \\ 1, & \text{otherwise} \end{cases} \tag{18}$$

where $\mathrm{Y}_1 := \sigma\sqrt{r-2}$. The original and positive part JS estimators are illustrated in Fig. 3 (a).

We propose a new local minimax self-weight estimation method that uses a direct bound with a specific upper-bound value, as follows:

$$p_i^{\mathrm{LMM-DB}} := \min(p_i^{\mathrm{LJS}_+}, p_i^{\max}). \tag{19}$$

This estimator is minimax under certain conditions that can be derived using Baranchik's minimax estimator theorem. According to this theorem, this operation can be interpreted as follows:

$$c(\|\mathbf{s}\|) = \begin{cases} \dfrac{\|\mathbf{s}\|^2}{\sigma^2(r-2)}, & 0 \le \|\mathbf{s}\| \le \mathrm{Y}_1 \\ 1, & \mathrm{Y}_1 < \|\mathbf{s}\| \le \mathrm{Y}_2 \\ \dfrac{\|\mathbf{s}\|^2(1-p^{\max})}{\sigma^2(r-2)}, & \mathrm{Y}_2 < \|\mathbf{s}\| \end{cases} \tag{20}$$

where $\mathrm{Y}_2 := \sigma\sqrt{(r-2)/(1-p^{\max})}$. We call this a local minimax self-weight estimator using direct bound (LMM-DB), which is illustrated in Fig. 3 (b) where $\mathrm{Y}_4 := \sigma\sqrt{2(r-2)/(1-p^{\max})}$.

However, note that LMM-DB is not minimax for $\|\mathbf{s}\| > \mathrm{Y}_4$. Fortunately, $\|\mathbf{s}\| = \|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|$ can be limited by adjusting the smoothing parameter $h$ by making it smaller so that all $\|\mathbf{s}\| \le \mathrm{Y}_4$ and $c\left(\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\|\right) \le 2$. Then, the LMM-DB becomes "practically" a minimax estimator. Let us denote the maximum $h$ that satisfies $\|\mathbf{s}\| \le \mathrm{Y}_4$ as $h^{\max}$.

In this case, a question can be raised: will the optimal value for $h$ fall into the range of $h$ that satisfies $\|\mathbf{s}\| \le \mathrm{Y}_4$? Interestingly, our simulations with many natural images showed that the optimal smoothing parameter $h^*$ based on the true images is very close to $h^{\max}$. This is because the LMM-DB yielded $p^{\max} \to 1$ so that $\mathrm{Y}_2 \to \infty$, and almost all $\left\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\right\|$ were less than or equal to $\mathrm{Y}_4$. Therefore, $p_i^{\mathrm{LMM-DB}}$ is "practically" a minimax value based on Baranchik's theorem for many natural images. Moreover, the LMM-DB method may provide a way to choose the optimal global smoothing parameter value $h$ without knowing the underlying true image. We empirically investigate this issue in Section V.

### C. Local Minimax Self-Weight Estimation With Re-Parametrization

The LMM-DB algorithm set $p$ to be the same $p^{\max}$ for a wide range of $\left\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\right\|$ values. We now propose another new method, called the local minimax self-weight estimation with reparametrization (LMM-RP) method, that assigns different $p$ values for different $\left\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\right\|$.

We reparametrized $p_i$ in (7) in the following way:

$$\begin{aligned} \hat{x}_i &= \hat{z}_i (p_i/p_i^{\max}) p_i^{\max}(y_i - \hat{z}_i) \\ &= \hat{z}_i + p_i^{\mathrm{T}}(y_i^{\mathrm{T}} - \hat{z}_i^{\mathrm{T}}) \qquad (21) \\ &= (1-p_i^{\max})\hat{z}_i + \hat{z}_i^{\mathrm{T}} + p_i^{\mathrm{T}}(y_i^{\mathrm{T}} - \hat{z}_i^{\mathrm{T}}) \qquad (22) \end{aligned}$$

where $\hat{z}_i^{\mathrm{T}} = p_i^{\max}\hat{z}_i$, $y_i^{\mathrm{T}} = p_i^{\max} y_i$, and

$$p_i^{\mathrm{T}} = \frac{1}{p_i^{\max}} \frac{w_{i,i}}{W_i + w_{i,i}}. \tag{23}$$

Note that for $0 \le w_{i,i} \le w_{i,i}^{\max}$, $p_i^{\mathrm{T}}$ is an increasing function of $w_{i,i}$ and the range of $p_i^{\mathrm{T}}$ is $0 \le p_i^{\mathrm{T}} \le 1$. We propose to use the positive part of the JS estimator to estimate the reparametrized $p_i^{\mathrm{T}}$, as follows:

$$\begin{aligned} p_i^{\mathrm{T,LJS}_+} &= \left[1 - \frac{(|\mathbf{B}|-2)\,(p_i^{\max})^2\sigma^2}{\left\|\mathbf{B}_i\mathbf{y}^{\mathrm{T}} - \mathbf{B}_i\hat{\mathbf{z}}^{\mathrm{T}}\right\|^2}\right]_+ \\ &= \left[1 - \frac{(|\mathbf{B}|-2)\,\sigma^2}{\left\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\right\|^2}\right]_+ = p_i^{\mathrm{LJS}_+}. \qquad (24) \end{aligned}$$

This method is equivalent to using a multiplicative factor $p_i^{\max}$ for the original JS shrinkage (9):

$$\hat{x}_i^{\mathrm{LMM-RP}} = (1 - p_i^{\mathrm{LMM-RP}})\hat{z}_i + p_i^{\mathrm{LMM-RP}} y_i \tag{25}$$

where

$$p_i^{\mathrm{LMM-RP}} = p_i^{\max}\left[1 - \frac{(|\mathbf{B}|-2)\,\sigma^2}{\left\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\right\|^2}\right]_+. \tag{26}$$

This proposed LMM-RP estimator is not dominant when estimating $x_i$, but rather is dominant when estimating $p_i^{\max}x_i$, as shown in (22). Thus, the positive part JS estimator does not guarantee that the LMM-RP is dominant.

Baranchik's minimax estimation theorem can be used to analyze the LMM-RP estimator as follows:

$$c\left(\|\mathbf{s}\|\right) = \begin{cases} \dfrac{\|\mathbf{s}\|^2}{\sigma^2(r-2)}, & 0 \le \|\mathbf{s}\| \le \Upsilon_1 \\ \dfrac{\|\mathbf{s}\|^2(1-p^{\max})}{\sigma^2(r-2)} + p^{\max}, & \Upsilon_1 < \|\mathbf{s}\| \end{cases} \tag{27}$$

where if $\|\mathbf{s}\|$ is $\Upsilon_3 := \sigma\sqrt{(2-p^{\max})(r-2)/(1-p^{\max})}$, then $c\left(\|\mathbf{s}\|\right) = 2$. The LMM-RP method is also illustrated in Fig. 3 (b), and is minimax if $\|\mathbf{s}\| \le \Upsilon_3$. The global smoothing parameter $h$ can be adjusted so that this condition is satisfied for different images. As in the case of the LMM-DB, it turns out that the optimal global smoothing parameter $h^*$ and the upper bound $h$ that satisfies $\|\mathbf{s}\| \le \Upsilon_3$ are also very close to each other when the LMM-RP method is applied to many natural images. Therefore, the LMM-RP method is "practically" a minimax. The following table summarizes the LJS self-weight estimation method and our proposed LMM-based self-weight estimation methods.

**Summary of Self-Weight Estimation Methods**

$\mathbf{LJS_+}$ [33] :

$$p_i^{\mathrm{LJS}_+} = \left[1 - (|\mathbf{B}|-2)\,\sigma^2 / \left\|\mathbf{B}_i\mathbf{y} - \mathbf{B}_i\hat{\mathbf{z}}\right\|^2\right]_+$$
$$\hat{x}_i^{\mathrm{LJS}_+} = (1 - p_i^{\mathrm{LJS}_+})\hat{z}_i + p_i^{\mathrm{LJS}_+} y_i$$

$\mathbf{LMM-DB}$ :

$$p_i^{\mathrm{LMM-DB}} = \min(p_i^{\mathrm{LJS}_+}, p_i^{\max})$$
$$\hat{x}_i^{\mathrm{LMM-DB}} = (1 - p_i^{\mathrm{LMM-DB}})\hat{z}_i + p_i^{\mathrm{LMM-DB}} y_i$$

$\mathbf{LMM-RP}$ :

$$p_i^{\mathrm{LMM-RP}} = p_i^{\mathrm{LJS}_+} p_i^{\max}$$
$$\hat{x}_i^{\mathrm{LMM-RP}} = (1 - p_i^{\mathrm{LMM-RP}})\hat{z}_i + p_i^{\mathrm{LMM-RP}} y_i$$

## V. Simulation Results

### A. Simulation Setup

Ten natural images[1] (cameraman, lena, montage, house, pepper, barbara, boat, hill, couple, fingerprint) and five images from the SUN database[2] (abbey, airplane cabin, airport terminal, alley, amphitheater) were used in our study as noise-free images (128 × 128, 256 × 256, or 512 × 512 pixels, 8 bits). A real patient MRI (512 × 512 pixels, 8 bits) that was acquired and processed under institutional review board (IRB) approved protocols was also used. White Gaussian noise was added to each input image with various standard deviations $\sigma \in \{10, 20, 40, 60\}$.

All algorithms were implemented using MATLAB R2015b (The Mathworks, Inc., Natick, MA, USA). The patch size and search window size of the NLM filter were chosen to be 7 × 7 and 31 × 31, respectively, which were the same as those used in [33]. Both the state-of-the-art LJS algorithm and the proposed algorithms were tested using $B = 1, \cdots, 9$ where $|\mathbf{B}| = (2B+1)^2 > 3$.

The global smoothing parameter $h$ was chosen empirically to yield the best PSNR:

$$\mathrm{PSNR}\left(\hat{\mathbf{x}}\right) = 10\log_{10}\frac{255^2}{\left\|\hat{\mathbf{x}} - \mathbf{x}\right\|^2/N}, \tag{28}$$

where $N$ is the size of the image. In addition to the PSNR, the mean bias vs. the mean variance trade-off curves were used as performance measures for the different smoothing parameter values $h$:

$$\overline{\mathrm{bias}^2} = \frac{1}{N}\sum_{i=1}^{N}(\bar{x}_i - x_i)^2, \tag{29}$$

$$\overline{\mathrm{var}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{k-1}\sum_{j=1}^{k}\left(\hat{x}_{ij} - \bar{x}_i\right)^2, \tag{30}$$

where $k$ is the number of realizations (20 in our simulation), $\hat{x}_{ij}$ is the $j$th estimation at the $i$th pixel, and $\bar{x}_i$ is the mean of $\hat{x}_{ij}$, as given by:

$$\bar{x}_i = \frac{1}{k}\sum_{j=1}^{k}\hat{x}_{ij}.$$

A visual quality assessment was also performed.

### B. Performance Studies Using the PSNR

In order to estimate values of $p_i$ for a fixed neighborhood size $B$, the optimal NLM smoothing parameter $h^*$ was determined such that the PSNR was maximized. In our proposed methods, the two maximum self-weights in (14) and (15) were used. The LMM-DB and LMM-RP methods given by (14) are denoted $\mathrm{LMM-DB}^{\mathrm{one}}$ and $\mathrm{LMM-RP}^{\mathrm{one}}$, while the LMM-DB and LMM-RP methods given by (15) are denoted $\mathrm{LMM-DB}^{\mathrm{stein}}$ and $\mathrm{LMM-RP}^{\mathrm{stein}}$. Table I summarizes the quantitative PSNR results for the 16 images with 4 different noise levels. When $B = 7$, our proposed LMM-DB and LMM-RP methods based on Baranchik's minimax estimator yielded much better PSNR results than did setting the self-weight to one in the classical NLM method [4], and comparable PSNR values to the LJS method based on the JS estimator [33]. When $B = 2$, our proposed LMM-DB and LMM-RP methods yielded better PSNR values than did the LJS.

For the five examples of lena, house, peppers, barbara, boat with $\sigma = 20$, PSNRs of our proposed methods (global smoothing parameter and fixed neighborhoods, but adaptive self-weight) were 0.72 ∼ 0.97 dB better than classical NLM. In [10], it is reported that for the same five examples with the same level of noise, the work of Kervrann et al. (fixed self-weight, but local smoothing parameters and adaptive neighborhoods) yielded 0.99 ∼ 1.55 dB better PSNR than classical NLM. Self-weights, local smoothing parameter, neighborhoods size are important factors in the NLM filter to determine output image quality.

[1] Available online at: http://www.cs.tut.fi/~foi/GCF-BM3D/BM3D_images.zip as the date of 16 Nov. 2015.

[2] Available online at: http://vision.princeton.edu/projects/2010/SUN/ as the date of 16 Sep. 2016.

TABLE I

PSNR (dB) SUMMARY (MEAN ± STANDARD DEVIATION) FOR VARIOUS NATURAL IMAGES

| | | | B = 2 | | | | | B = 7 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | σ | Classical NLM | LJS$_+$ | LMM-DB$^{one}$ | LMM-RP$^{one}$ | LMM-DB$^{stein}$ | LMM-RP$^{stein}$ | LJS$_+$ | LMM-DB$^{one}$ | LMM-RP$^{one}$ | LMM-DB$^{stein}$ | LMM-RP$^{stein}$ |
| *cameraman* | *10* | 32.42 ± 0.034 | 33.12 ± 0.031 | **33.32 ± 0.030** | 33.29 ± 0.029 | 33.17 ± 0.029 | 33.04 ± 0.030 | 32.98 ± 0.035 | 33.10 ± 0.036 | 33.05 ± 0.037 | 32.98 ± 0.038 | 32.85 ± 0.040 |
| | *20* | 28.48 ± 0.052 | 29.12 ± 0.052 | **29.46 ± 0.056** | 29.29 ± 0.056 | 29.27 ± 0.060 | 28.97 ± 0.052 | 29.32 ± 0.062 | 29.34 ± 0.059 | 29.04 ± 0.054 | 29.11 ± 0.058 | 28.80 ± 0.049 |
| | *40* | 25.35 ± 0.059 | 25.39 ± 0.083 | 25.89 ± 0.075 | **26.11 ± 0.082** | 26.08 ± 0.087 | 25.74 ± 0.079 | 25.98 ± 0.073 | 25.98 ± 0.073 | 25.94 ± 0.080 | 25.96 ± 0.083 | 25.63 ± 0.076 |
| | *60* | 23.19 ± 0.065 | 22.88 ± 0.062 | 23.39 ± 0.055 | **23.74 ± 0.065** | 23.67 ± 0.069 | 23.54 ± 0.061 | 23.68 ± 0.068 | 23.69 ± 0.068 | 23.62 ± 0.071 | 23.63 ± 0.070 | 23.54 ± 0.059 |
| *lena* | *10* | 33.90 ± 0.018 | 34.52 ± 0.017 | 34.74 ± 0.017 | 34.81 ± 0.018 | 34.77 ± 0.017 | 34.69 ± 0.017 | 34.82 ± 0.020 | **34.83 ± 0.020** | 34.80 ± 0.019 | 34.76 ± 0.018 | 34.63 ± 0.018 |
| | *20* | 30.78 ± 0.031 | 30.90 ± 0.023 | 31.25 ± 0.033 | 31.50 ± 0.032 | **31.51 ± 0.032** | 31.31 ± 0.029 | **31.51 ± 0.031** | **31.51 ± 0.031** | 31.45 ± 0.028 | 31.48 ± 0.031 | 31.27 ± 0.029 |
| | *40* | 27.64 ± 0.032 | 26.94 ± 0.028 | 27.74 ± 0.033 | 28.08 ± 0.029 | 28.08 ± 0.029 | 28.06 ± 0.028 | **28.10 ± 0.028** | **28.10 ± 0.028** | 28.07 ± 0.028 | 28.08 ± 0.029 | 28.06 ± 0.028 |
| | *60* | 25.60 ± 0.052 | 24.38 ± 0.040 | 25.66 ± 0.051 | **26.02 ± 0.052** | 26.01 ± 0.053 | **26.02 ± 0.054** | 26.01 ± 0.051 | 26.01 ± 0.051 | **26.02 ± 0.054** | **26.02 ± 0.053** | **26.02 ± 0.054** |
| *montage* | *10* | 34.68 ± 0.045 | 35.19 ± 0.043 | 35.60 ± 0.042 | **35.67 ± 0.039** | 35.65 ± 0.042 | 35.55 ± 0.045 | 35.12 ± 0.049 | 35.39 ± 0.046 | 35.38 ± 0.042 | 35.46 ± 0.045 | 35.34 ± 0.047 |
| | *20* | 30.35 ± 0.088 | 30.74 ± 0.062 | 31.29 ± 0.067 | 31.38 ± 0.070 | **31.40 ± 0.078** | 31.06 ± 0.068 | 31.00 ± 0.076 | 31.13 ± 0.073 | 31.07 ± 0.068 | 31.18 ± 0.068 | 30.81 ± 0.063 |
| | *40* | 26.24 ± 0.063 | 26.30 ± 0.072 | 26.98 ± 0.070 | 27.29 ± 0.063 | **27.30 ± 0.064** | 27.20 ± 0.061 | 27.01 ± 0.052 | 27.03 ± 0.052 | 27.04 ± 0.055 | 27.08 ± 0.053 | 27.00 ± 0.053 |
| | *60* | 23.76 ± 0.104 | 23.48 ± 0.116 | 24.16 ± 0.113 | **24.61 ± 0.115** | 24.60 ± 0.106 | 24.38 ± 0.115 | 24.33 ± 0.092 | 24.33 ± 0.092 | 24.36 ± 0.090 | 24.37 ± 0.088 | 24.16 ± 0.095 |
| *house* | *10* | 34.57 ± 0.038 | 35.02 ± 0.039 | 35.36 ± 0.041 | **35.38 ± 0.039** | 35.34 ± 0.043 | 35.29 ± 0.046 | 35.31 ± 0.042 | 35.32 ± 0.043 | 35.25 ± 0.044 | 35.21 ± 0.047 | 35.12 ± 0.045 |
| | *20* | 31.43 ± 0.063 | 31.54 ± 0.048 | 32.13 ± 0.050 | **32.39 ± 0.048** | **32.39 ± 0.050** | 32.19 ± 0.067 | 32.30 ± 0.052 | 32.31 ± 0.054 | 32.26 ± 0.058 | 32.30 ± 0.056 | 32.10 ± 0.068 |
| | *40* | 27.62 ± 0.044 | 27.18 ± 0.038 | 27.84 ± 0.049 | **28.37 ± 0.037** | **28.37 ± 0.039** | 28.33 ± 0.045 | 28.35 ± 0.041 | 28.35 ± 0.041 | 28.34 ± 0.045 | 28.35 ± 0.042 | 28.33 ± 0.045 |
| | *60* | 25.01 ± 0.092 | 24.24 ± 0.087 | 25.17 ± 0.098 | 25.65 ± 0.095 | 25.65 ± 0.087 | 25.65 ± 0.088 | 25.65 ± 0.088 | 25.65 ± 0.088 | 25.65 ± 0.087 | **25.66 ± 0.087** | 25.65 ± 0.088 |
| *peppers* | *10* | 32.62 ± 0.056 | 33.37 ± 0.042 | 33.53 ± 0.048 | **33.56 ± 0.049** | 33.49 ± 0.051 | 33.39 ± 0.050 | 33.28 ± 0.043 | 33.37 ± 0.040 | 33.35 ± 0.042 | 33.33 ± 0.042 | 33.17 ± 0.042 |
| | *20* | 28.94 ± 0.031 | 29.54 ± 0.029 | 29.78 ± 0.040 | **29.88 ± 0.038** | 29.86 ± 0.028 | 29.51 ± 0.027 | 29.77 ± 0.027 | 29.79 ± 0.027 | 29.70 ± 0.026 | 29.73 ± 0.024 | 29.34 ± 0.033 |
| | *40* | 25.31 ± 0.050 | 25.50 ± 0.057 | 25.67 ± 0.041 | **26.12 ± 0.049** | 26.11 ± 0.054 | 25.97 ± 0.055 | 26.08 ± 0.054 | 26.08 ± 0.054 | 26.04 ± 0.056 | 26.05 ± 0.054 | 25.95 ± 0.055 |
| | *60* | 22.99 ± 0.048 | 22.95 ± 0.091 | 23.18 ± 0.061 | 23.80 ± 0.067 | 23.80 ± 0.071 | 23.78 ± 0.075 | **23.81 ± 0.070** | **23.81 ± 0.070** | 23.79 ± 0.074 | 23.80 ± 0.073 | 23.78 ± 0.075 |
| *barbara* | *10* | 32.93 ± 0.026 | 33.50 ± 0.018 | 33.66 ± 0.020 | 33.70 ± 0.021 | 33.66 ± 0.022 | 33.53 ± 0.020 | 33.72 ± 0.017 | **33.74 ± 0.017** | 33.69 ± 0.017 | 33.66 ± 0.017 | 33.44 ± 0.017 |
| | *20* | 29.36 ± 0.032 | 29.83 ± 0.029 | 29.96 ± 0.032 | 30.23 ± 0.030 | **30.27 ± 0.028** | 30.04 ± 0.029 | **30.27 ± 0.029** | **30.27 ± 0.028** | 30.19 ± 0.026 | 30.24 ± 0.027 | 30.00 ± 0.030 |
| | *40* | 25.68 ± 0.047 | 25.78 ± 0.048 | 25.79 ± 0.047 | 26.46 ± 0.043 | 26.51 ± 0.040 | 26.51 ± 0.039 | **26.52 ± 0.040** | **26.52 ± 0.040** | 26.51 ± 0.039 | 26.51 ± 0.040 | 26.51 ± 0.039 |
| | *60* | 23.50 ± 0.032 | 23.17 ± 0.039 | 23.57 ± 0.034 | 24.13 ± 0.037 | 24.15 ± 0.035 | **24.16 ± 0.035** | 24.15 ± 0.036 | 24.15 ± 0.036 | **24.16 ± 0.035** | **24.16 ± 0.035** | **24.16 ± 0.035** |
| *boat* | *10* | 31.78 ± 0.015 | 32.73 ± 0.019 | 32.81 ± 0.018 | **32.82 ± 0.017** | 32.72 ± 0.015 | 32.61 ± 0.016 | 32.73 ± 0.018 | 32.75 ± 0.018 | 32.72 ± 0.017 | 32.65 ± 0.017 | 32.49 ± 0.017 |
| | *20* | 28.40 ± 0.017 | 29.14 ± 0.017 | 29.23 ± 0.019 | **29.37 ± 0.015** | 29.34 ± 0.015 | 29.05 ± 0.015 | 29.30 ± 0.018 | 29.30 ± 0.018 | 29.25 ± 0.017 | 29.27 ± 0.017 | 28.95 ± 0.018 |
| | *40* | 21.95 ± 0.053 | 25.45 ± 0.021 | 25.45 ± 0.021 | **26.01 ± 0.016** | 25.99 ± 0.016 | 25.92 ± 0.014 | 25.98 ± 0.012 | 25.98 ± 0.012 | 25.95 ± 0.013 | 25.96 ± 0.012 | 25.92 ± 0.014 |
| | *60* | 23.64 ± 0.025 | 23.11 ± 0.028 | 23.72 ± 0.026 | **24.01 ± 0.026** | **24.01 ± 0.025** | 23.99 ± 0.025 | **24.01 ± 0.025** | **24.01 ± 0.025** | 24.00 ± 0.025 | 24.00 ± 0.025 | 23.99 ± 0.025 |
| *hill* | *10* | 31.87 ± 0.029 | 32.63 ± 0.020 | 32.67 ± 0.019 | **32.71 ± 0.018** | 32.61 ± 0.018 | 32.47 ± 0.014 | 32.67 ± 0.016 | 32.67 ± 0.016 | 32.64 ± 0.016 | 32.55 ± 0.015 | 32.34 ± 0.013 |
| | *20* | 28.82 ± 0.022 | 29.23 ± 0.031 | 29.23 ± 0.031 | **29.48 ± 0.024** | **29.48 ± 0.023** | 29.29 ± 0.023 | 29.45 ± 0.021 | 29.45 ± 0.021 | 29.41 ± 0.023 | 29.42 ± 0.022 | 29.25 ± 0.024 |
| | *40* | 25.91 ± 0.022 | 25.70 ± 0.026 | 25.98 ± 0.024 | 26.36 ± 0.024 | **26.38 ± 0.022** | 26.37 ± 0.022 | **26.38 ± 0.022** | **26.38 ± 0.022** | 26.37 ± 0.022 | 26.38 ± 0.022 | 26.37 ± 0.022 |
| | *60* | 24.25 ± 0.017 | 23.45 ± 0.013 | 24.32 ± 0.018 | **24.60 ± 0.024** | 24.59 ± 0.024 | **24.60 ± 0.024** | 24.59 ± 0.022 | 24.59 ± 0.022 | **24.60 ± 0.024** | **24.60 ± 0.024** | **24.60 ± 0.024** |
| *couple* | *10* | 31.80 ± 0.014 | 32.76 ± 0.009 | 32.81 ± 0.009 | **32.85 ± 0.010** | 32.80 ± 0.011 | 32.71 ± 0.011 | 32.76 ± 0.013 | 32.77 ± 0.013 | 32.75 ± 0.012 | 32.72 ± 0.011 | 32.59 ± 0.012 |
| | *20* | 28.14 ± 0.023 | 28.93 ± 0.023 | 28.93 ± 0.023 | **29.16 ± 0.028** | **29.16 ± 0.029** | 28.86 ± 0.030 | 29.11 ± 0.028 | 29.11 ± 0.028 | 29.07 ± 0.029 | 29.08 ± 0.029 | 28.76 ± 0.030 |
| | *40* | 24.93 ± 0.035 | 25.03 ± 0.026 | 25.05 ± 0.026 | 25.49 ± 0.026 | **25.50 ± 0.030** | 25.44 ± 0.032 | 25.48 ± 0.031 | 25.48 ± 0.031 | 25.47 ± 0.030 | 25.47 ± 0.030 | 25.43 ± 0.032 |
| | *60* | 23.25 ± 0.037 | 22.76 ± 0.043 | 23.29 ± 0.036 | 23.59 ± 0.045 | **23.60 ± 0.044** | 23.59 ± 0.044 | **23.60 ± 0.044** | **23.60 ± 0.044** | 23.59 ± 0.044 | **23.60 ± 0.044** | 23.59 ± 0.044 |
| *fingerprint* | *10* | 30.27 ± 0.017 | 30.87 ± 0.015 | 30.87 ± 0.016 | 30.84 ± 0.016 | 30.80 ± 0.017 | 30.57 ± 0.016 | **30.88 ± 0.018** | **30.88 ± 0.019** | 30.83 ± 0.019 | 30.81 ± 0.020 | 30.50 ± 0.018 |
| | *20* | 26.64 ± 0.010 | 27.06 ± 0.014 | 27.06 ± 0.014 | 27.04 ± 0.014 | **27.12 ± 0.012** | 26.72 ± 0.013 | 27.10 ± 0.012 | 27.10 ± 0.012 | 26.93 ± 0.013 | 27.05 ± 0.013 | 26.70 ± 0.012 |
| | *40* | 23.20 ± 0.018 | 23.68 ± 0.024 | 23.68 ± 0.024 | 23.96 ± 0.023 | **24.06 ± 0.022** | 24.05 ± 0.023 | 24.05 ± 0.022 | 24.05 ± 0.022 | 24.05 ± 0.022 | 24.05 ± 0.022 | 24.05 ± 0.023 |
| | *60* | 20.93 ± 0.034 | 21.44 ± 0.029 | 21.44 ± 0.029 | 21.85 ± 0.041 | 21.97 ± 0.037 | **21.98 ± 0.037** | **21.98 ± 0.037** | **21.98 ± 0.037** | **21.98 ± 0.037** | **21.98 ± 0.037** | **21.98 ± 0.037** |
| *MRI* | *10* | 40.06 ± 0.043 | 39.19 ± 0.040 | 40.81 ± 0.033 | **40.89 ± 0.032** | 40.83 ± 0.034 | 40.71 ± 0.029 | 40.79 ± 0.040 | 40.85 ± 0.038 | 40.83 ± 0.037 | 40.81 ± 0.038 | 40.61 ± 0.032 |
| | *20* | 36.14 ± 0.067 | 34.47 ± 0.047 | 36.57 ± 0.062 | 36.70 ± 0.063 | 36.74 ± 0.065 | 36.60 ± 0.068 | 36.74 ± 0.063 | **36.77 ± 0.064** | 36.64 ± 0.068 | 36.72 ± 0.066 | 36.59 ± 0.067 |
| | *40* | 32.22 ± 0.067 | 29.33 ± 0.055 | 32.31 ± 0.071 | 32.53 ± 0.069 | 32.53 ± 0.069 | 32.52 ± 0.069 | 32.49 ± 0.064 | **32.54 ± 0.070** | 32.52 ± 0.069 | **32.54 ± 0.069** | 32.52 ± 0.069 |
| | *60* | 29.57 ± 0.056 | 26.13 ± 0.058 | 29.68 ± 0.057 | **29.88 ± 0.056** | **29.88 ± 0.056** | **29.88 ± 0.056** | 29.76 ± 0.060 | 29.86 ± 0.059 | **29.88 ± 0.056** | **29.88 ± 0.056** | **29.88 ± 0.056** |
| *abbey* | *10* | 29.31 ± 0.035 | **29.96 ± 0.029** | **29.96 ± 0.029** | 29.86 ± 0.031 | 29.87 ± 0.033 | 29.38 ± 0.030 | 29.92 ± 0.028 | 29.92 ± 0.028 | 29.83 ± 0.032 | 29.83 ± 0.033 | 29.34 ± 0.030 |
| | *20* | 25.53 ± 0.034 | **25.91 ± 0.036** | **25.91 ± 0.036** | 25.87 ± 0.033 | 25.80 ± 0.034 | 25.27 ± 0.034 | 25.87 ± 0.034 | 25.87 ± 0.034 | 25.72 ± 0.032 | 25.71 ± 0.034 | 25.16 ± 0.033 |
| | *40* | 22.85 ± 0.036 | 22.94 ± 0.031 | 22.94 ± 0.031 | 23.10 ± 0.030 | **23.14 ± 0.024** | 23.10 ± 0.024 | 23.13 ± 0.025 | 23.13 ± 0.025 | 23.12 ± 0.025 | 23.12 ± 0.025 | 23.10 ± 0.025 |
| | *60* | 21.60 ± 0.034 | 21.30 ± 0.031 | 21.58 ± 0.032 | 21.83 ± 0.038 | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** | **21.85 ± 0.039** |
| *airplane cabin* | *10* | 31.42 ± 0.077 | 32.47 ± 0.088 | **32.54 ± 0.086** | 32.49 ± 0.085 | 32.50 ± 0.084 | 32.05 ± 0.082 | 32.36 ± 0.078 | 32.39 ± 0.077 | 32.30 ± 0.085 | 32.33 ± 0.084 | 31.78 ± 0.082 |
| | *20* | 27.52 ± 0.107 | 28.53 ± 0.110 | 28.60 ± 0.117 | **28.70 ± 0.130** | 28.58 ± 0.129 | 28.21 ± 0.119 | 28.57 ± 0.133 | 28.57 ± 0.133 | 28.45 ± 0.127 | 28.46 ± 0.127 | 28.11 ± 0.121 |
| | *40* | 24.62 ± 0.112 | 24.86 ± 0.124 | 24.91 ± 0.105 | 25.26 ± 0.113 | **25.28 ± 0.112** | 25.24 ± 0.121 | 25.25 ± 0.120 | 25.25 ± 0.120 | 25.25 ± 0.118 | 25.25 ± 0.117 | 25.24 ± 0.120 |
| | *60* | 22.89 ± 0.185 | 22.60 ± 0.176 | 22.97 ± 0.194 | 23.36 ± 0.206 | **23.39 ± 0.192** | 23.38 ± 0.193 | 23.38 ± 0.193 | 23.38 ± 0.193 | 23.38 ± 0.193 | 23.38 ± 0.193 | 23.38 ± 0.193 |
| *airport terminal* | *10* | 32.79 ± 0.042 | 33.41 ± 0.038 | **33.58 ± 0.036** | 33.51 ± 0.038 | **33.58 ± 0.043** | 33.09 ± 0.045 | 33.26 ± 0.041 | 33.34 ± 0.041 | 33.23 ± 0.044 | 33.37 ± 0.044 | 32.86 ± 0.040 |
| | *20* | 28.74 ± 0.071 | 29.58 ± 0.050 | 29.72 ± 0.044 | 29.95 ± 0.055 | **29.97 ± 0.054** | 29.67 ± 0.049 | 29.70 ± 0.054 | 29.70 ± 0.054 | 29.69 ± 0.058 | 29.74 ± 0.057 | 29.42 ± 0.047 |
| | *40* | 24.78 ± 0.060 | 25.21 ± 0.055 | 25.21 ± 0.055 | 25.70 ± 0.077 | **25.71 ± 0.077** | 25.55 ± 0.070 | 25.61 ± 0.074 | 25.61 ± 0.074 | 25.59 ± 0.074 | 25.60 ± 0.074 | 25.51 ± 0.076 |
| | *60* | 22.72 ± 0.061 | 22.59 ± 0.049 | 22.80 ± 0.063 | 23.22 ± 0.072 | **23.23 ± 0.073** | 23.20 ± 0.070 | 23.21 ± 0.070 | 23.21 ± 0.070 | 23.21 ± 0.071 | 23.21 ± 0.071 | 23.20 ± 0.070 |
| *alley* | *10* | 37.58 ± 0.075 | 37.65 ± 0.053 | 37.90 ± 0.070 | 38.44 ± 0.077 | **38.48 ± 0.076** | 38.42 ± 0.082 | 38.44 ± 0.079 | 38.44 ± 0.079 | 38.44 ± 0.079 | 38.45 ± 0.078 | 38.41 ± 0.081 |
| | *20* | 33.90 ± 0.087 | 33.35 ± 0.087 | 34.15 ± 0.102 | 34.59 ± 0.101 | 34.59 ± 0.101 | 34.59 ± 0.101 | 34.59 ± 0.101 | 34.59 ± 0.101 | **34.60 ± 0.101** | **34.60 ± 0.101** | 34.59 ± 0.102 |
| | *40* | 31.06 ± 0.056 | 28.67 ± 0.123 | 31.12 ± 0.057 | **31.27 ± 0.058** | 31.26 ± 0.059 | 31.26 ± 0.060 | 31.16 ± 0.077 | 31.22 ± 0.064 | 31.26 ± 0.059 | **31.27 ± 0.060** | 31.26 ± 0.060 |
| | *60* | 29.65 ± 0.101 | 25.84 ± 0.110 | 29.68 ± 0.101 | **29.77 ± 0.102** | 29.76 ± 0.102 | **29.77 ± 0.102** | 29.46 ± 0.116 | 29.72 ± 0.102 | **29.77 ± 0.102** | **29.77 ± 0.103** | **29.77 ± 0.102** |
| *amphitheater* | *10* | 32.27 ± 0.038 | 32.87 ± 0.038 | **33.02 ± 0.034** | 32.94 ± 0.035 | 32.90 ± 0.036 | 32.39 ± 0.050 | 32.94 ± 0.040 | 32.96 ± 0.037 | 32.83 ± 0.052 | 32.81 ± 0.049 | 32.23 ± 0.069 |
| | *20* | 28.70 ± 0.089 | 28.94 ± 0.061 | 29.09 ± 0.064 | **29.12 ± 0.071** | 29.02 ± 0.079 | 28.58 ± 0.078 | 29.11 ± 0.073 | 29.11 ± 0.073 | 28.94 ± 0.081 | 28.91 ± 0.079 | 28.52 ± 0.077 |
| | *40* | 25.88 ± 0.135 | 25.40 ± 0.093 | 25.90 ± 0.126 | 26.02 ± 0.157 | 26.02 ± 0.173 | 25.99 ± 0.175 | **26.06 ± 0.158** | **26.06 ± 0.158** | 26.01 ± 0.173 | 26.01 ± 0.172 | 25.99 ± 0.175 |
| | *60* | 24.62 ± 0.077 | 23.45 ± 0.159 | 24.64 ± 0.080 | 24.72 ± 0.081 | 24.72 ± 0.082 | 24.72 ± 0.082 | 24.72 ± 0.096 | **24.73 ± 0.093** | 24.72 ± 0.082 | 24.72 ± 0.082 | 24.72 ± 0.082 |

### *C. Performance Studies With Bias-Variance Trade-Off*

The bias-variance trade-off was investigated using many natural images. As shown in Fig. 1, a neighborhood size $B$ was used to estimate $p_i$ using the LJS method [33], and this was a significant factor when determining the bias. This tendency was also observed for the other different natural images, as illustrated in Fig. 4. Increasing $B$ in the LJS method moved the bias-variance trade-off curves in the bottom right direction, meaning that the bias increased and the variance decreased. However, the role of the smoothing parameter $h$ changed in the LJS method. Unlike in classical NLM method (see the NLM bias-variance curve in Fig. 1), increasing the smoothing parameter $h$ beyond a certain point in the LJS method did not further decrease the variance in any of the natural images that we tested. This is because increasing $h$ will also increases the $p_i$ values so that the resulting LJS estimator becomes closer to the noisy input image $y_i$ due to the lack of bounds for the self-weights.

Our proposed methods (LMM-DB, LMM-RP) yielded trade-off curves that have decreased variances for increasing values of the smoothing parameter $h$. Figure 5 shows the trade-off curves for the cameraman example for different methods (LMM-DB, LMM-RP), different neighborhood sizes ($B = 2, 7$), and different noise levels ($\sigma = 10, 40$). Our proposed methods yielded bias-variance curves that were less than or equal to those in the LJS method for fixed $B$ and $\sigma$. This tendency was also observed with other natural images, as illustrated in Fig. 4. It was important to choose appropriate

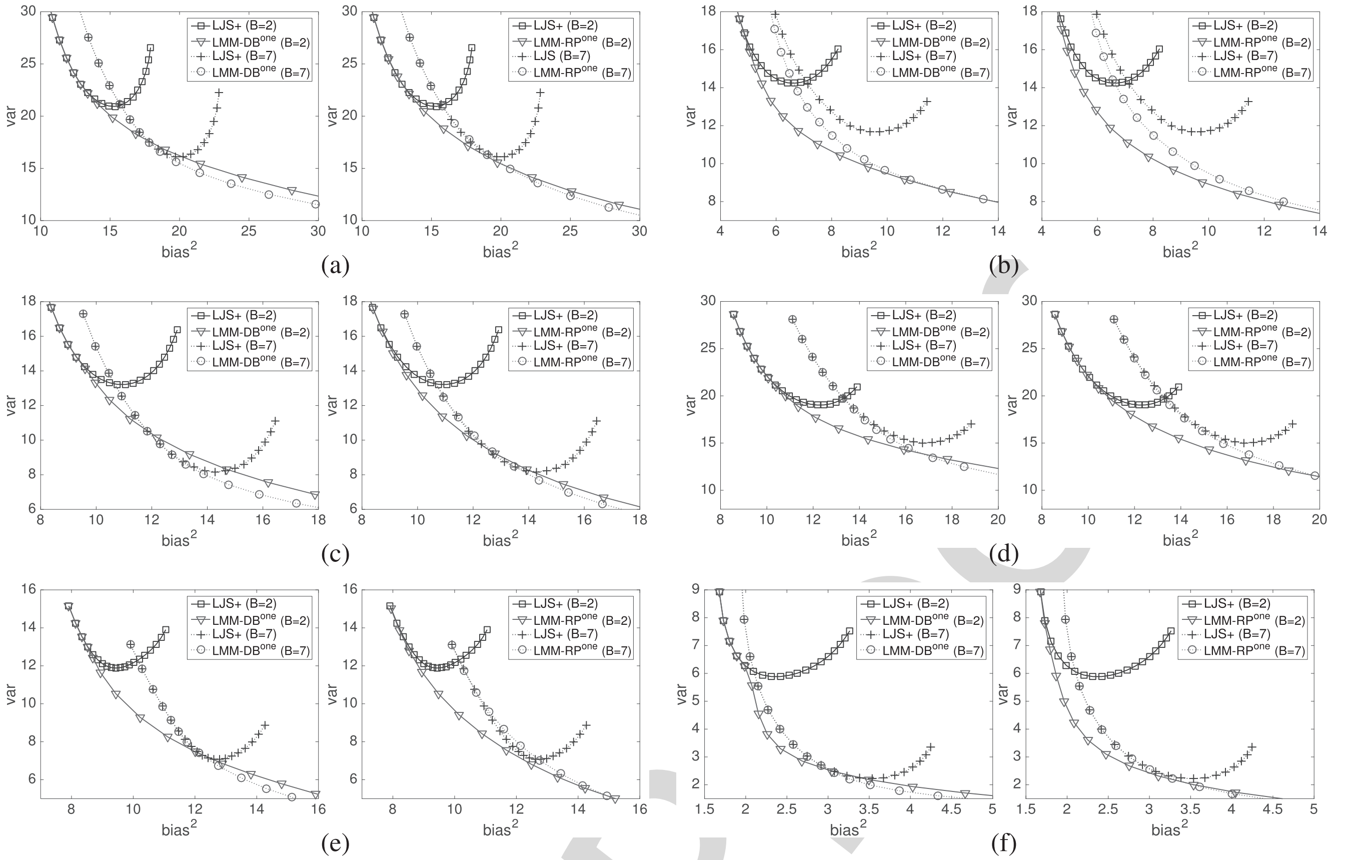



Fig. 4. Bias-variance curves for natural images using $\mathrm{LJS}_{+}$ [33] and our proposed $\mathrm{LMM-DB}^{\mathrm{one}}$ and $\mathrm{LMM-RP}^{\mathrm{one}}$ methods with a noise level of $\sigma = 10$. (a) *couple*. (b) *montage*. (c) *lena*. (d) *pepper*. (e) *house*. (f) *MRI*.

neighborhood sizes $B$ in order for the LJS method to obtain a certain level of bias, but our proposed methods were able to achieve that same level of bias by adjusting the smoothing parameter $h$, which was the same as in classical NLM. Based on our results, it appears that the use of LMM-RP has slightly more advantages than using LMM-DB in terms of the PSNR, as shown in Table I, and the bias-variance trade-off curves, as shown in Fig. 5, for high noise levels.

### D. Performance Studies With Visual Quality Assessment

The most important improvements in our proposed LMM-DB and LMM-RP methods when compared to the LJS method were achieved in terms of the visual quality. Figure 6 (a) shows the true cameraman image (left) and the noisy image (right) with a noise level of $\sigma = 10$. Figure 6 (b) presents the filtered images using the LJS method [33] with $B = 2$ and $B = 7$. Severe artifacts were observed in the background areas when using $B = 2$, and these artifacts were reduced when using $B = 7$. However, there were still some artifacts near the edges of objects. Our proposed LMM-DB and LMM-RP methods exhibited fewer image artifacts than were observed in the images processed using the LJS method for both $B = 2, 7$. This tendency was observed in many of the natural images, as shown in Fig. 7, especially in the high intensity flat areas. PSNR improvements in the LJS method were achieved with severe (when $B = 2$) or mild (when $B = 7$) artifacts; however, our proposed methods achieved both a high PSNR and significantly reduced visual artifacts. This ability to reduce the number of visual artifacts in a denoised image is important in some applications, such as diagnostic medical imaging.

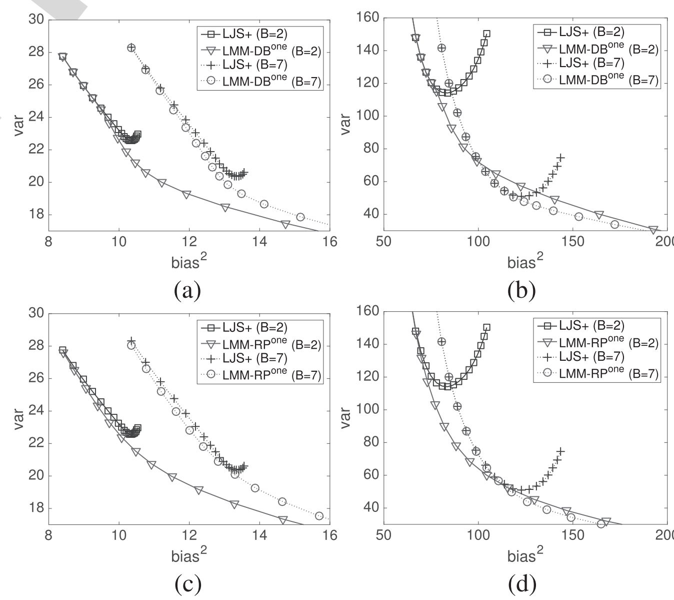



Fig. 5. Bias-variance curves for $\mathrm{LMM-DB}^{\mathrm{one}}$ and $\mathrm{LMM-RP}^{\mathrm{one}}$ for comparison with $\mathrm{LJS}_{+}$ for two neighborhood sizes $B = 2, 7$ and two noise levels $\sigma = 10, 40$. (a) $\mathrm{LMM-DB}^{\mathrm{one}}(\sigma = 10)$. (b) $\mathrm{LMM-DB}^{\mathrm{one}}(\sigma = 40)$. (c) $\mathrm{LMM-RP}^{\mathrm{one}}(\sigma = 10)$. (d) $\mathrm{LMM-RP}^{\mathrm{one}}(\sigma = 40)$.

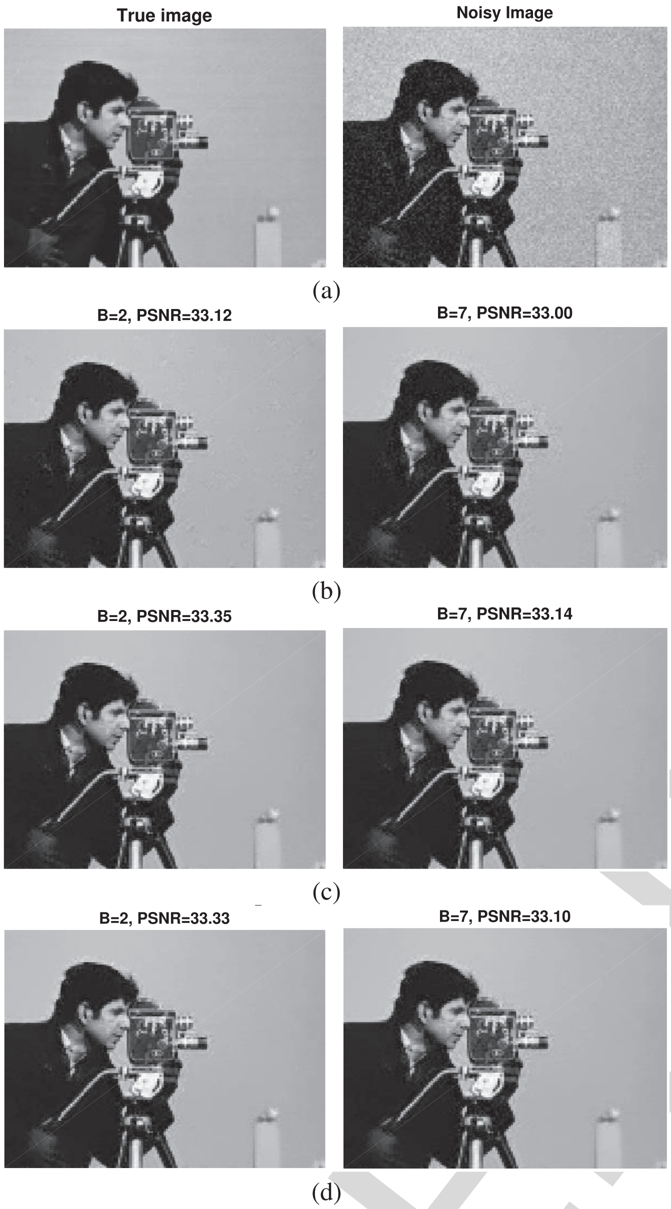



Fig. 6. True, noisy ($\sigma = 10$), and filtered images using $\mathrm{LJS}_+$ [33] , and the proposed $\mathrm{LMM-DB}^{\mathrm{one}}$ and $\mathrm{LMM-RP}^{\mathrm{one}}$. (a) True and noisy images ($\sigma = 10$). (b) $\mathrm{LJS}_+$ [33]. (c) Proposed $\mathrm{LMM-DB}^{\mathrm{one}}$. (d) Proposed $\mathrm{LMM-RP}^{\mathrm{one}}$.

### E. Maximum Self-Weights: One vs. Stein's

Two maximum self-weights were proposed for use: the value one in (14) that was proposed in [4], and Stein's in (15) that was proposed in [32]. Figure 8 shows that the $\mathrm{LMM-DB}^{\mathrm{one}}$ method yielded an improved bias-variance curve and PSNR than did the $\mathrm{LMM-DB}^{\mathrm{stein}}$ method when the noise levels were low. For high noise levels $\sigma = 40$, the $\mathrm{LMM-DB}^{\mathrm{stein}}$ method yielded an improved PSNR and bias-variance curve than did the $\mathrm{LMM-DB}^{\mathrm{one}}$ method. However, these differences were not significant, as also illustrated in terms of the PSNR in Table I. In terms of the visual quality, no significant differences were observed between the two methods.

TABLE II

PERCENTAGE (%) OF $c(\|\mathbf{s}\|)$ THAT EXCEED 2 USING $\mathrm{LMM-DB}$ AND $\mathrm{LMM-RP}$ METHODS, $\sigma = 10$, B = 2

| | | $\sigma = 10$ | $\sigma = 20$ | $\sigma = 40$ | $\sigma = 60$ |
|---|---|---|---|---|---|
| *cameraman* | $\mathrm{LMM\text{-}DB}^{\mathrm{one}}$ | 0.32 | 0.04 | 0.03 | 0.05 |
| | $\mathrm{LMM\text{-}DB}^{\mathrm{stein}}$ | 0.85 | 0.66 | 0.21 | 0.18 |
| *fingerprint* | $\mathrm{LMM\text{-}DB}^{\mathrm{one}}$ | 0.00 | 0.00 | 0.00 | 0.00 |
| | $\mathrm{LMM\text{-}DB}^{\mathrm{stein}}$ | 0.30 | 0.13 | 0.09 | 0.02 |
| *MRI* | $\mathrm{LMM\text{-}DB}^{\mathrm{one}}$ | 0.10 | 0.05 | 0.10 | 0.13 |
| | $\mathrm{LMM\text{-}DB}^{\mathrm{stein}}$ | 0.18 | 0.16 | 0.16 | 0.16 |

| | | $\sigma = 10$ | $\sigma = 20$ | $\sigma = 40$ | $\sigma = 60$ |
|---|---|---|---|---|---|
| *cameraman* | $\mathrm{LMM\text{-}RP}^{\mathrm{one}}$ | 0.25 | 0.04 | 0.01 | 0.00 |
| | $\mathrm{LMM\text{-}RP}^{\mathrm{stein}}$ | 1.07 | 0.90 | 0.20 | 0.22 |
| *fingerprint* | $\mathrm{LMM\text{-}RP}^{\mathrm{one}}$ | 0.01 | 0.00 | 0.00 | 0.00 |
| | $\mathrm{LMM\text{-}RP}^{\mathrm{stein}}$ | 0.27 | 0.19 | 0.13 | 0.03 |
| *MRI* | $\mathrm{LMM\text{-}RP}^{\mathrm{one}}$ | 0.09 | 0.05 | 0.07 | 0.09 |
| | $\mathrm{LMM\text{-}RP}^{\mathrm{stein}}$ | 0.23 | 0.20 | 0.16 | 0.13 |

### F. "Practical" Minimax Estimator

The proposed LMM-DB and LMM-RP methods are minimax with respect to $\|\mathbf{s}\| \leq \Upsilon_4$ and $\|\mathbf{s}\| \leq \Upsilon_3$, respectively, as shown in Fig. 3. However, these conditions impose upper bounds for the smoothing parameters $h$ and the optimal $h^*$, which means that the smoothing parameter values that yield the best PSNR may not be achievable. We empirically investigated this issue using many natural images.

Table II shows the ratio (percentage unit) of the number of pixels for which $c(\|\mathbf{s}\|) > 2$ to the total number of pixels in the cameraman, fingerprint, and MRI images when the optimal $h^*$ for the highest PSNR was chosen based on the true images for the proposed LMM-DB and LMM-RP methods. For most of the pixels, the LMM-DB and LMM-RP values were minimax. The relationship between the percentage of pixels with $c(\|\mathbf{s}\|) > 2$ and the root mean squared error (RMSE) is illustrated in Fig. 9 for the cameraman and MRI images. Surprisingly, the optimal global smoothing parameters $h$ for the lowest RMSE point (or the highest PSNR) of the LMM-DB and LMM-RP methods are very close to the smoothing parameters $h$ such that the percentage of $c(\|\mathbf{s}\|) > 2$ is 0.1%. This phenomenon was not only observed in these two images. As shown in Table III, the pixel percentage of $c(\|\mathbf{s}\|) > 2$ that do not require knowledge of the true image can still determine smoothing parameters that are able to yield comparable PSNR values to the best PSNR values obtained by using the optimal smoothing parameters calculated based on knowledge of the true image. This was observed in all of the natural images used in our simulations, with different noise levels, and when $B = 2$ was used. However, the criteria of using the pixel percentage of $c(\|\mathbf{s}\|) > 2$ did not work very well for $B = 7$ in our simulations. These criteria can be potentially used when choosing a global smoothing parameter with our proposed methods as a heuristic approach without knowing the true image.

### G. Computation Time for Algorithms

Table IV reports the computation time of the proposed methods in comparison with the classical NLM and $\mathrm{LJS}_+$.

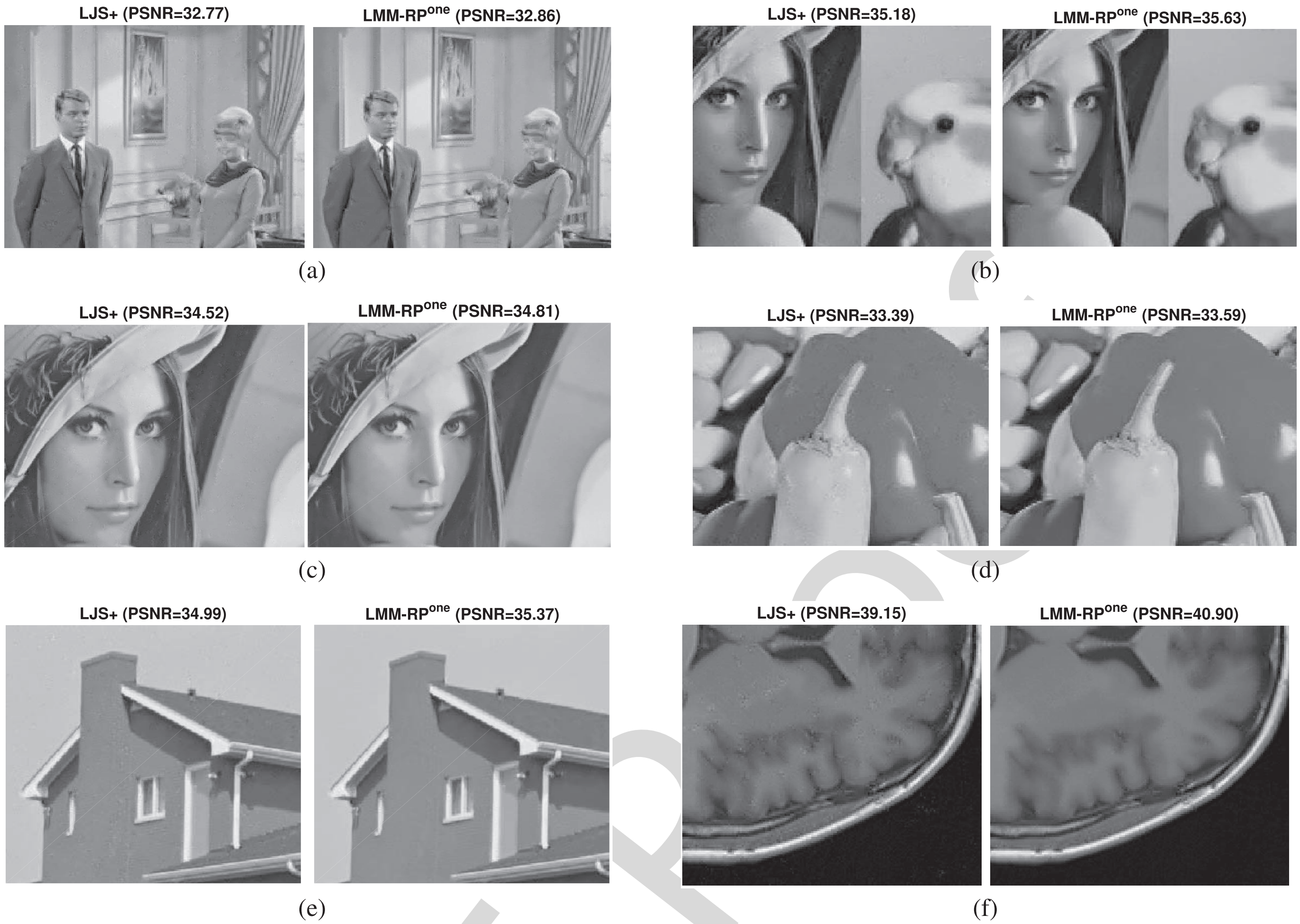



Fig. 7. Filtered results using $\text{LJS}_+$ [33] and the proposed $\text{LJS} - \text{RP}^{\text{one}}$ method with a noise level of $\sigma = 10$ and neighborhood size B = 2. (a) *couple*. (b) *montage*. (c) *lena*. (d) *pepper*. (e) *house*. (f) *MRI*.

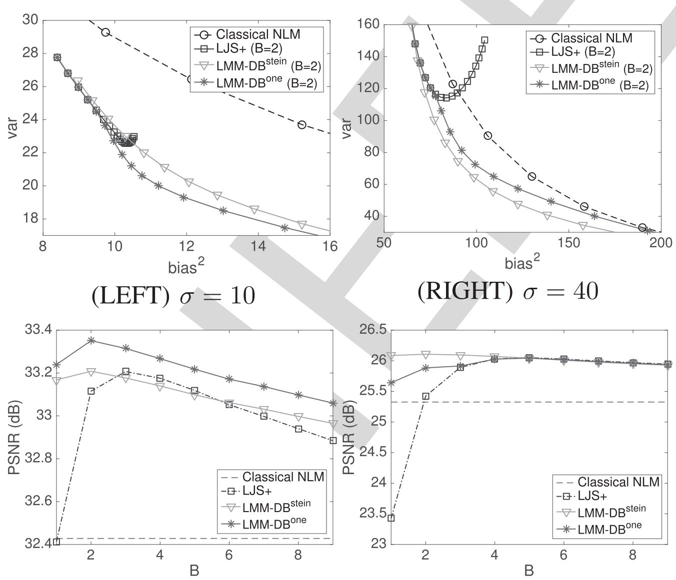



Fig. 8. Bias-variance curves and PSNR vs. varying neighborhood sizes ($B$) using classical NLM (only in the PSNR figure), LJS, and the proposed $\text{LMM} - \text{DB}^{\text{stein}}$ vs. $\text{LMM} - \text{DB}^{\text{one}}$ for the cameraman example.

We used 8 threads (Intel Core i7 2.8 GHz) when computing the patch distances for all methods. The local block size was $B = 2$, the patch size was $7 \times 7$, and the window size was $31 \times 31$. All parameters were fixed for all of results presented in this section. Adjusting these parameters can greatly reduce the running time. For example, setting $B = 4$, the patch size to $5 \times 5$, and the window size to $13 \times 13$ reduces the computation time of the proposed methods to 0.60, 1.12, and 2.91 seconds (s) for $128^2$, $256^2$, and $512^2$ images, respectively. However, analytically, the classical NLM requires $3|\mathbf{P}||\mathbf{\Omega}| + 4|\mathbf{\Omega}| - 1$ operations per pixel where $|\mathbf{\Omega}|$ is the number of elements in $\mathbf{\Omega}_i$ and $\text{LJS}_+$ requires $3|\mathbf{P}||\mathbf{\Omega}| + 4|\mathbf{\Omega}| + 3|\mathbf{B}| + 5$ operations per pixel. It is reported in [33] that the additional operations for $\text{LJS}_+$ ($3|\mathbf{B}| + 6$ operations) were negligible compared to the NLM filtering computation ($3|\mathbf{P}||\mathbf{\Omega}| + 4|\mathbf{\Omega}| - 1$ operations). Analytically, the additional computation for $\text{LMM} - \text{DB}$ and $\text{LMM} - \text{RP}$ is $3|\mathbf{B}| + 7$ operations, which is almost the same as the additional computation for $\text{LJS}_+$. Therefore, further implementation optimization is possible by exploiting the redundant computation of the patch distances for the minimax estimator and NLM weights.

## VI. Discussion

The classical NLM method was a significant work in image denoising [4], and required the determination of two important parameters for good denoising performance: a smoothing parameter and a self-weight value. The LJS method proposed

TABLE III

THE PSNR VALUES (dB) OF THE PROPOSED METHODS WITH B = 2 WHEN CHOOSING THE SMOOTHING PARAMETER SO AS TO YIELD THE HIGHEST PSNR USING THE TRUE IMAGE (TRUE), AND WHEN CHOOSING THE SMOOTHING PARAMETER SO AS TO YIELD THE PERCENTAGE OF $c(\|\mathbf{s}\|) > 2$ TO BE 0.1% (ESTIMATED) FOR DIFFERENT NOISE LEVELS

| | | LMM-DB $^{one}$ | | LMM-RP $^{one}$ | |
|---|---|---|---|---|---|
| | σ | TRUE | ESTIMATED | TRUE | ESTIMATED |
| ***cameraman*** | *10* | 33.35 | 33.32 | 33.22 | 33.30 |
| | *20* | 29.47 | 29.47 | 29.30 | 29.45 |
| | *40* | 25.91 | 25.90 | 26.16 | 26.01 |
| | *60* | 23.43 | 23.43 | 23.78 | 23.60 |
| ***lena*** | *10* | 34.72 | 34.74 | 34.73 | 34.78 |
| | *20* | 31.22 | 31.20 | 31.47 | 31.37 |
| | *40* | 27.74 | 27.74 | 28.07 | 27.88 |
| | *60* | 25.60 | 25.63 | 25.95 | 25.82 |
| ***montage*** | *10* | 35.55 | 35.56 | 35.51 | 35.34 |
| | *20* | 31.24 | 31.20 | 31.33 | 31.32 |
| | *40* | 26.99 | 26.98 | 27.26 | 27.13 |
| | *60* | 24.07 | 23.97 | 24.54 | 24.16 |
| ***house*** | *10* | 35.32 | 35.35 | 35.30 | 35.37 |
| | *20* | 32.00 | 31.97 | 32.30 | 32.20 |
| | *40* | 27.82 | 27.79 | 28.35 | 28.06 |
| | *60* | 25.23 | 25.23 | 25.75 | 25.37 |
| ***peppers*** | *10* | 33.56 | 33.54 | 33.48 | 33.59 |
| | *20* | 29.81 | 29.80 | 29.91 | 29.95 |
| | *40* | 25.71 | 25.67 | 26.16 | 25.89 |
| | *60* | 23.15 | 23.02 | 23.81 | 23.24 |
| ***barbara*** | *10* | 33.67 | 33.68 | 33.62 | 33.73 |
| | *20* | 29.94 | 29.82 | 30.23 | 30.03 |
| | *40* | 25.79 | 25.69 | 26.47 | 25.92 |
| | *60* | 23.56 | 23.48 | 24.14 | 23.67 |
| ***boat*** | *10* | 32.80 | 32.80 | 32.73 | 32.79 |
| | *20* | 29.22 | 29.18 | 29.36 | 29.22 |
| | *40* | 25.44 | 25.60 | 25.99 | 25.75 |
| | *60* | 23.76 | 23.76 | 24.05 | 23.91 |
| ***hill*** | *10* | 32.66 | 32.64 | 32.60 | 32.61 |
| | *20* | 29.24 | 29.17 | 29.49 | 29.30 |
| | *40* | 26.01 | 25.99 | 26.38 | 26.05 |
| | *60* | 24.35 | 24.35 | 24.63 | 24.45 |
| ***couple*** | *10* | 32.82 | 32.80 | 32.79 | 32.81 |
| | *20* | 28.89 | 28.70 | 29.12 | 28.85 |
| | *40* | 25.08 | 25.05 | 25.52 | 25.10 |
| | *60* | 23.26 | 23.19 | 23.56 | 23.28 |
| ***fingerprint*** | *10* | 30.86 | 30.84 | 30.66 | 30.80 |
| | *20* | 27.07 | 26.86 | 27.05 | 26.96 |
| | *40* | 23.69 | 23.24 | 23.96 | 23.38 |
| | *60* | 21.47 | 20.77 | 21.92 | 21.02 |
| ***MRI*** | *10* | 40.83 | 40.83 | 40.77 | 40.90 |
| | *20* | 36.59 | 36.60 | 36.71 | 36.74 |
| | *40* | 32.36 | 32.36 | 32.58 | 32.56 |
| | *60* | 29.64 | 29.63 | 29.83 | 29.80 |
| ***abbey*** | *10* | 30.01 | 29.98 | 29.98 | 29.94 |
| | *20* | 25.89 | 25.74 | 25.91 | 25.60 |
| | *40* | 22.96 | 22.80 | 23.03 | 22.81 |
| | *60* | 21.58 | 21.58 | 21.71 | 21.65 |
| ***airplane cabin*** | *10* | 32.47 | 32.45 | 32.50 | 32.49 |
| | *20* | 28.51 | 28.35 | 28.61 | 28.45 |
| | *40* | 24.74 | 24.68 | 25.00 | 24.81 |
| | *60* | 23.13 | 23.04 | 23.34 | 23.30 |
| ***airport terminal*** | *10* | 33.64 | 33.64 | 33.68 | 33.68 |
| | *20* | 29.63 | 29.53 | 29.83 | 29.63 |
| | *40* | 25.19 | 25.04 | 25.51 | 25.07 |
| | *60* | 22.95 | 22.94 | 23.23 | 23.08 |
| ***alley*** | *10* | 37.94 | 37.87 | 38.28 | 38.06 |
| | *20* | 34.15 | 34.10 | 34.44 | 34.25 |
| | *40* | 31.06 | 31.03 | 31.19 | 31.19 |
| | *60* | 29.53 | 29.19 | 29.59 | 29.49 |
| ***amphitheater*** | *10* | 33.00 | 32.97 | 33.00 | 32.94 |
| | *20* | 29.18 | 29.08 | 29.31 | 29.13 |
| | *40* | 25.90 | 25.90 | 25.98 | 25.95 |
| | *60* | 24.54 | 24.51 | 24.59 | 24.59 |

by Wu *et al.* [33] developed a state-of-the-art method for self-weight determination using JS estimation [37] and yielded superior results in terms of the PSNR compared to the other existing methods. However, since the LJS method did not impose an upper bound for self-weight estimation, the bias could no longer be controlled by the smoothing parameter, which resulted in visual quality degradation. Our proposed methods based on the Baranchik's minimax theorem [34] yielded comparable PSNR results to the state-of-the-art LJS method. By imposing upper bounds for the self-weights,

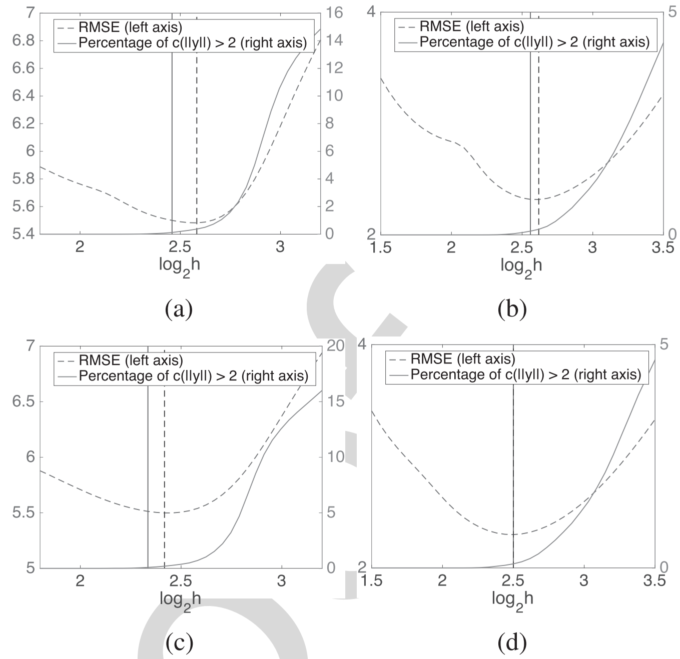



Fig. 9. Comparison plots of the RMSE vs. the smoothing parameter $h$ and the percentage of $c(\|s\|) > 2$ vs. the same smoothing parameter when using LMM-DB and LMM-RP with B = 2 and $\sigma = 10$. (a) cameraman $\mathrm{LMM-DB^{one}}$. (b) MRI $\mathrm{LMM-DB^{one}}$. (c) cameraman $\mathrm{LMM-RP^{one}}$. (d) MRI $\mathrm{LMM-RP^{one}}$.

TABLE IV

EXECUTION TIME (S) COMPARISON. THIS WILL VARY WITH PARAMETER SELECTION

| Image size | Classical NLM | $\mathrm{LJS_+}$ | LMM-DB $^{one}$ | LMM-RP $^{one}$ |
|---|---|---|---|---|
| **128*128** | 0.65 | 0.89 | 0.90 | 0.90 |
| **256*256** | 1.57 | 2.37 | 2.39 | 2.38 |
| **512*512** | 4.90 | 7.14 | 7.19 | 7.18 |

the bias-variance trade-off was able to be controlled by a smoothing parameter, and substantial visual artifact reduction was achieved.

The focus of this article was self-weight parameter selection in the classical NLM filter with theoretical justification. As discussed in the Introduction, there are other factors that affect the performance of NLM based filters, and we expect that our proposed methods would not be able to achieve state-of-the-art denoising performance if there were no other optimizations performed except the self-weights. Indeed, our proposed methods with one patch size (non-adaptive neighborhood) and one global smoothing parameter were not able to achieve the level of denoising performance of the state-of-the-art denoising methods such as BM3D [6]. However, when our proposed methods have incorporated some of the other factors into the NLM filters, such as local smoothing parameters and adaptive neighborhoods [10], they have great potential to achieve significantly improved denoising performance.

The minimax property of our proposed methods depends on the choice of smoothing parameters. When using sufficiently small smoothing parameters, the LMM-DB and LMM-RP methods are "practically" minimax according to Baranchik's theorem [34]. However, when large smoothing parameters are used, there may be some pixels that are not minimax for self-weight estimation. More empirical investigation showed that the optimal global smoothing parameter $h$ that yielded the best PSNR only resulted in a very small portion of the pixels that did not have minimax self-weight estimators. In fact, this can be used as a useful heuristic when choosing a good smoothing parameter since testing the minimax properties of our proposed methods does not require the true image. More theoretical analysis for this observation, or a statistical analysis using many natural images as shown in [41], are potential extensions of this work. Therefore, our proposed methods do not only provide an optimal way to determine self-weights, but also provide a heuristic way to determine a good smoothing parameter.

## VII. Conclusion

We proposed two methods, LMM-DB, LMM-RP, to determine the self-weights of NLM filters that are "practically" minimax, and this methods yielded a comparable PSNR, better bias-variance trade-offs, and reduced visual quality artifacts when compared to the results obtained using the state-of-the-art LJS method. Our methods also provide a potentially useful heuristic way to determine a global smoothing parameter without knowledge of the original image.

## References


[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Mumbai, India, Jan. 1998, pp. 839–846.

[2] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.

[3] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

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

[5] S. P. Awate and R. T. Whitaker, "Unsupervised, information-theoretic, adaptive image filtering for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 364–376, Mar. 2006.

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

[7] C. Louchet and L. Moisan, "Total variation as a local filter," *SIAM J. Imag. Sci.*, vol. 4, no. 2, pp. 651–694, Jan. 2011.

[8] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2012.

[9] D. Barash, "Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 844–847, Jun. 2002.

[10] 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.

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

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

[13] 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.

[14] Z. Ji, Q. Chen, Q.-S. Sun, and D.-S. Xia, "A moment-based nonlocal-means algorithm for image denoising," *Inf. Process. Lett.*, vol. 109, nos. 23–24, pp. 1238–1244, Nov. 2009.

[15] S. Grewenig, S. Zimmer, and J. Weickert, "Rotationally invariant similarity measures for nonlocal image denoising," *J. Vis. Commun. Image Represent.*, vol. 22, no. 2, pp. 117–130, Feb. 2011.

[16] R. Yan, L. Shao, S. D. Cvetković, and J. Klijn, "Improved nonlocal means based on pre-classification and invariant block matching," *J. Display Technol.*, vol. 8, no. 4, pp. 212–218, Apr. 2012.

[17] C. A. Deledalle, V. Duval, and J. Salmon, "Non-local methods with shape-adaptive patches (NLM-SAP)," *J. Math. Imag. Vis.*, vol. 43, no. 2, pp. 103–120, Jun. 2012.

[18] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1635–1649, Apr. 2012.

[19] S. Y. Chun, J. A. Fessler, and Y. K. Dewaraja, "Post-reconstruction nonlocal means filtering methods using CT side information for quantitative SPECT," *Phys. Med. Biol.*, vol. 58, no. 17, pp. 6225–6240, 2013.

[20] C. Chan, R. Fulton, R. Barnett, D. Feng, and S. Meikle, "Postreconstruction nonlocal means filtering of whole-body PET with an anatomical prior," *IEEE Trans. Med. Imag.*, vol. 33, no. 3, pp. 636–650, Mar. 2014.

[21] 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.

[22] D. Van De Ville and M. Kocher, "Nonlocal means with dimensionality reduction and SURE-based parameter selection," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2683–2690, Sep. 2011.

[23] C.-A. Deledalle, F. Tupin, and L. Denis, "Poisson NL means: Unsupervised non local means for Poisson noise," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 801–804.

[24] Y. Wu, B. H. Tracey, P. Natarajan, and J. P. Noonan, "Fast blockwise SURE shrinkage for image denoising," *Signal Process.*, vol. 103, pp. 45–59, Oct. 2014.

[25] B. Goossens, Q. Luong, A. Pižurica, and W. Philips, "An improved non-local denoising algorithm," in *Proc. Int. Workshop Local Non-Local Approx. Image Process.*, Lausanne, Switzerland, 2008, pp. 143–156.

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

[27] C. Kervrann, J. Boulanger, and P. Coupé, "Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal," in *Scale Space and Variational Methods in Computer Vision*. Berlin, Germany: Springer, 2007, pp. 520–532.

[28] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, "Probabilistic non-local means," *IEEE Signal Process. Lett.*, vol. 20, no. 8, pp. 763–766, Aug. 2013.

[29] V. Dore 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.

[30] T. Brox and D. Cremers, "Iterated nonlocal means for texture restoration," in *Scale Space and Variational Methods in Computer Vision*. Berlin, Germany: Springer, 2007, pp. 13–24.

[31] 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 Approx. Image Process.*, 2008, pp. 135–142.

[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] Y. Wu, B. Tracey, P. Natarajan, and J. P. Noonan, "James–Stein type center pixel weights for non-local means image denoising," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 411–414, Apr. 2013.

[34] A. J. Baranchik, "A family of minimax estimators of the mean of a multivariate normal distribution," *Ann. Math. Statist.*, vol. 41, no. 2, pp. 642–645, 1970.

[35] M. P. Nguyen and S. Y. Chun, "Center pixel weight estimation for non-local means filtering using local James-Stein estimator with bounded self-weights," in *Proc. IEEE 13th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2016, pp. 82–85.

[36] C. Stein, "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution," in *Proc. 3rd Berkeley Symp. Math. Statist. Probab.*, vol. 1. 1956, pp. 197–206.

[37] W. James and C. Stein, "Estimation with quadratic loss," in *Proc. 4th Berkeley Symp. Math. Statist. Probab.*, vol. 1. 1961, pp. 361–379.

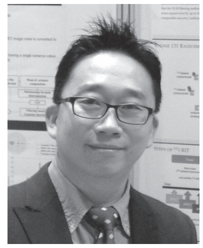
[38] A. J. Baranchik, "Multiple regression and estimation of the mean of a multivariate normal distribution," Dept. Statist., Stanford Univ., Stanford, CA, USA, Tech. Rep. 51, May 1964.

[39] B. Efron and C. Morris, "Families of minimax estimators of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 4, no. 1, pp. 11–21, 1976.

[40] W. E. Strawderman, "Proper Bayes minimax estimators of the multivariate normal mean vector for the case of common unknown variances," *Ann. Statist.*, vol. 1, no. 6, pp. 1189–1194, 1973.

[41] X. Li, H. He, R. Wang, and D. Tao, "Single image superresolution via directional group sparsity and directional features," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2874–2888, Sep. 2015.

**Minh Phuong Nguyen** received the B.S.E. degree in biomedical electronics from the School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam, in 2013, and the M.S. degree in electrical engineering from the Ulsan National Institute of Science and Technology, Ulsan, South Korea. He is currently pursuing the Ph.D. degree with the Delft University of Technology, The Netherlands. His research interests include image reconstruction and image filtering for SPECT, image registration between histology images and MRI, and electrocardiogram signal processing for authentication tasks.



**Se Young Chun** (M'10) received the B.S.E. degree from the School of Electrical Engineering, Seoul National University, Seoul, South Korea, in 1999, and the M.S.E. degree in electrical engineering systems, the M.S. degree in mathematics, and the Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, MI, USA, in 2005 and 2009, respectively.

From 2003 to 2005, he was a recipient of the two year scholarship from the Ministry of Information and Communications, South Korea. From 2009 to 2011, he was a Research Fellow with Massachusetts General Hospital and Harvard Medical School, Boston, MA, USA, and from 2011 to 2013, he was a Research Fellow with Electrical Engineering and Computer Science and Radiology, University of Michigan. Since 2013, he has been an Assistant Professor with the School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan, South Korea. His research interests include image reconstruction, image denoising, image registration for multi-modal imaging systems, motion compensated image reconstruction, and statistical bio signal processing and biometrics.

Dr. Chun was a recipient of the 2010 Society of Nuclear Medicine Computer and Instrumentation Young Investigator Award, 2nd place. He was also a recipient of the 2015 Bruce Hasegawa Young Investigator Medical Imaging Science Award from the IEEE Nuclear and Plasma Sciences Society. He is a member of IEEE Computational Imaging Special Interest Group and an Editorial Board Member of *Nuclear Medicine and Molecular Imaging* (Springer).