Patch-Based Near-Optimal Image Denoising

Priyam Chatterjee, *Student Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

***Abstract*—In this paper, we propose a denoising method motivated by our previous analysis of the performance bounds for image denoising. Insights from that study are used here to derive a high-performance practical denoising algorithm. We propose a patch-based Wiener filter that exploits patch redundancy for image denoising. Our framework uses both geometrically and photometrically similar patches to estimate the different filter parameters. We describe how these parameters can be accurately estimated directly from the input noisy image. Our denoising approach, designed for near-optimal performance (in the mean-squared error sense), has a sound statistical foundation that is analyzed in detail. The performance of our approach is experimentally verified on a variety of images and noise levels. The results presented here demonstrate that our proposed method is on par or exceeding the current state of the art, both visually and quantitatively.**

***Index Terms*—Denoising bounds, image clustering, image denoising, linear minimum mean-squared-error (LMMSE) estimator, Wiener filter.**

I. Introduction

In RECENT years, images and videos have become integral parts of our lives. Applications now range from the casual documentation of events and visual communication to the more serious surveillance and medical fields. This has led to an ever-increasing demand for accurate and visually pleasing images. However, images captured by modern cameras are invariably corrupted by noise [3]. With increasing pixel resolution but more or less the same aperture size, noise suppression has become more relevant. While advances in optics and hardware try to mitigate such undesirable effects, software-based denoising approaches are more popular as they are usually device independent and widely applicable. In the last decade, many such methods have been proposed, leading to considerable improvement in denoising performance. In [1] and [2], we studied the problem from an estimation theory perspective to quantify the fundamental limits of denoising. The insights gained from that study are applied to develop a theoretically sound denoising method in this paper.

Manuscript received January 27, 2011; revised May 19, 2011 and September 16, 2011; accepted September 17, 2011. Date of publication October 19, 2011; date of current version March 21, 2012. This work was supported in part by the Air Force Office of Scientific Research under Grant FA9550-07-1-0365 and in part by the National Science Foundation under Grant CCF-1016018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Pascal Frossard.

P. Chatterjee was with the Department of Electrical Engineering, University of California, Santa Cruz, CA 95064 USA. He is now with Pelican Imaging Corporation, Mountain View, CA 94041 USA (e-mail: priyam@soe.ucsc.edu).

P. Milanfar is with the Department of Electrical Engineering, University of California, Santa Cruz, CA 95064 USA (e-mail: milanfar@soe.ucsc.edu).

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

Digital Object Identifier 10.1109/TIP.2011.2172799

The challenge of any image denoising algorithm is to suppress noise while producing sharp images without loss of finer details. The first modern adaptive method to successfully address these contradictory goals can be attributed to Tomasi *et al.* [4], where the authors proposed a generalization of the SUSAN filter [5], which itself was an extension of the Yaroslavky filter [6]. The authors there proposed denoising by weighted averaging pixels similar in intensity within a local neighborhood. Under strong noise, identifying such similar pixels can be challenging. In [7], Takeda *et al.* proposed a signal-dependent steering kernel regression (SKR) framework for denoising. This method proved to be much more robust under strong noise. A patch-based generalization of the bilateral filter [4] was proposed in [8] and [9], where the concept of locality was extended to the entire image. Although the results there were encouraging, the true potential for this nonlocal means (NLM) method was only realized in [10] and [11]. Another patch redundancy-based framework, i.e., BM3D [12], adopts a hybrid approach of grouping similar patches and performing collaborative filtering in some transform [e.g., discrete cosine transform (DCT)] domain. It ranks among the best performing methods that define the current state of the art.

A significantly different approach to denoising was introduced in K-SVD [13]. Building on the notion of image patches being sparse representable [14], Elad *et al.* proposed a greedy approach for dictionary learning tuned for denoising. In [15], we proposed a hybrid approach (K-LLD) that bridged such dictionary-based approaches with the regression-based frameworks [4], [7], [8], [10]. The motivation there was that similar patches shared similar *subdictionaries*, and such subdictionaries could be used for better image modeling. A similar observation was exploited in the form of a nonlocal sparse model (NLSM) [16] to improve performance of the K-SVD [13] framework.

The dictionary-based methods provide implicit modeling for natural images. More explicit models have also been used for denoising. In [17], Joshi *et al.* address the problem of denoising color images by explicitly modeling each pixel as a combination of two colors, the basis colors themselves being estimated from a local neighborhood. Denoising is achieved by enforcing such a model. The use of local principal components as bases for denoising was proposed in [18]. Zhang *et al.* [19] later refined this idea with a local pixel grouping mechanism such that the principal component bases are estimated from only similar patches within a neighborhood. Another model-based approach using Markov random fields (MRFs) as a field of experts (FOE) was applied to denoising natural images in [20], where the parameters for the model are learned from example images. Liu *et al.* [21] proposed a method where, in addition to a locally affine signal model, the noise level is also estimated from the input image, leading to a practical denoising method.

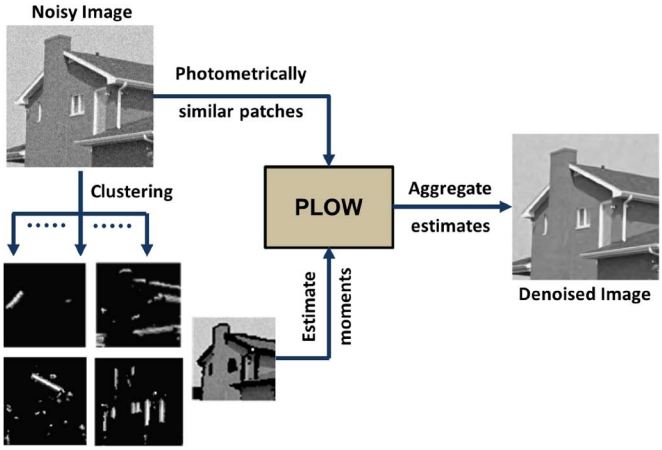


Fig. 1. Outline of our proposed PLOW filtering method.

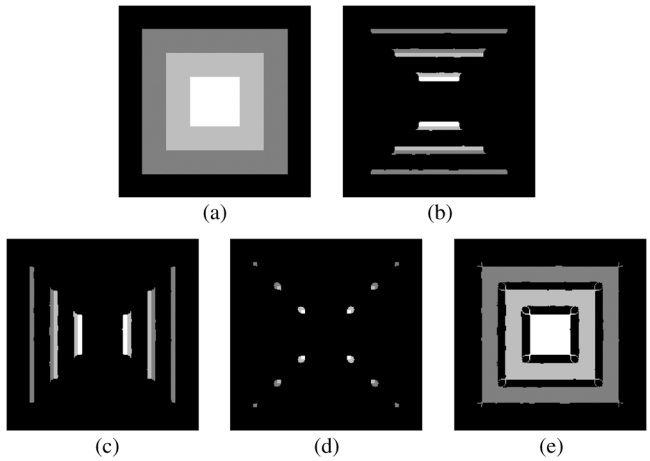


Fig. 2. Clustering of a simple image based on geometric similarity. Note how pixels in any particular cluster can have quite different intensities but similar geometric structure (edge, corner, flat regions, etc.) (a) Box image. (b) Cluster 1. (c) Cluster 2. (d) Cluster 3. (e) Cluster 4.

While most of the aforementioned approaches work in the spatial domain, a vast section of image denoising literature is devoted to transform domain methods. The main motivation in such methods is that, in the transform (e.g., DCT, wavelets, etc.) domain, it is possible to separate image and noise components, and denoising can be performed through the shrinkage of the transform coefficients. In [22], Chang *et al.* showed that, using a spatially adaptive threshold parameter, along with the overcomplete wavelet basis, denoising performance can be considerably improved. Another wavelet-domain method in [23] was considered the state of the art until recently. There, the authors perform denoising by modeling the wavelet coefficients of images as mixtures of Gaussians. Enforcing such a model for noisy images leads to considerable denoising. In [24], an additional global model for natural images in the form of homogeneous Gaussian MRFs was used to improve the performance considerably. In [25], Luisier *et al.* proposed a denoising method aimed at reducing the estimated mean-squared error (MSE) through wavelet thresholding.

In this paper, we propose a new denoising filter motivated by our statistical analysis of the performance bounds for patch-based methods [1], [2]. The contributions of our paper are as follows: We design a patch-based statistically motivated redundancy exploiting the Wiener filter, where the parameters of the method are learned from *both* geometrically and photometrically similar patches. As will be clear from our discussions in the next section, our method is formulated to approach the performance bounds for patch-based denoising. As a side note, we also show that the NLM filter [8], [9] is an approximation of the optimal filter (in the MSE sense) obtained if one ignores the geometric structure of image patches.

Although extensively used for denoising, the Wiener filter is usually used in conjunction with some transform basis. For example, the collaborative Wiener filter used in BM3D [12] works in the DCT domain where an estimate of the ground truth (signal-to-noise ratio) is obtained through an initial filtering of the image. Our spatial domain method is motivated by our analysis of the image denoising bounds [1], [2]. In our framework, graphically illustrated in Fig. 1, we develop a locally optimal Wiener filter where the parameters are learned from both geometrically and photometrically similar patches. For this, the noisy image is first segmented into regions of similar geometric structure, as shown in Fig. 2. The mean and the covariance of the patches within each cluster are then estimated. Next, for each patch, we identify photometrically similar patches and compute weights based on their similarity to the reference patch. These parameters are then used to perform denoising patchwise. To reduce artifacts, image patches are selected to have some degree of overlap (shared pixels) with their neighbors. A final aggregation step is then used to optimally fuse the multiple estimates for pixels lying on the patch overlaps to form the denoised image.

The remainder of this paper is organized as follows: In Section II, we motivate our algorithm and develop the theoretical formulation of our method. The details of how the parameters are estimated from the noisy image are presented in Section III. In Section IV, we present experimental validation of our method and compare it, visually and quantitatively, to some recently proposed popular denoising methods. Finally, we conclude our paper with some directions for future research in Section V.

II. PLOW Filter

A. Motivation

In [1], we analyzed the performance bounds for the problem of image denoising. This was done from an estimation theory point of view, where we seek to estimate the pixel intensity z_i at each location i from its noisy observation, i.e.,

$$y_i = z_i + \eta_i, \quad i = 1, \ldots, M. \tag{1}$$

Here, η_i is assumed to be independent and identically distributed (i.i.d.), and M is the total number of pixels in the image. In our paper, we specifically considered patch-based methods, where the observation model can be posed as

$$\mathbf{y}_i = \mathbf{z}_i + \boldsymbol{\eta}_i \tag{2}$$

with $\mathbf{y}_i \in \Re^n$ representing the vectorized $\sqrt{n} \times \sqrt{n}$ patch centered at i. Using a Bayesian Cramér–Rao bound [26]–[28] anal-

ysis, we showed that the MSE of denoising (estimating) any given patch in the image is bounded from below by

$$E\left[||\mathbf{z}_i - \widehat{\mathbf{z}}_i||^2\right] \geqslant \mathrm{Tr}\left[\left(\mathbf{J}_i + \mathbf{C}_{\mathbf{z}}^{-1}\right)^{-1}\right] \tag{3}$$

where $\widehat{\mathbf{z}}_i \in \Re^n$ is the estimate of $\mathbf{z}_i$, $\mathbf{J}_i \in \Re^{n\times n}$ is the Fisher information matrix (FIM), $\mathbf{C}_{\mathbf{z}} \in \Re^{n\times n}$ is the patch covariance matrix, and $||.||$ denotes the ℓ_2 norm. This covariance matrix captures the complexity of the patches and is estimated from all the geometrically similar patches present in the given image. Fig. 2 illustrates what we mean by geometric similarity, where it can be seen that each cluster groups together patches containing flat regions, edges in the horizontal or vertical directions, and corners of the simulated box image. Note that such grouping is done irrespective of the actual patch intensities. This is justified for intensity-independent noise when denoising performance is dictated by the complexity of patches, rather than their actual intensities.

The FIM, on the other hand, is influenced by the noise characteristics. When additive white Gaussian noise (WGN) is considered, the FIM takes the following form:

$$\mathbf{J}_i = N_i \frac{\mathbf{I}}{\sigma^2} \tag{4}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix, σ is the noise standard deviation, and N_i is the patch redundancy measured as the number of patches $\mathbf{z}_j$ within the latent image that are photometrically similar to a given patch $\mathbf{z}_i$. We define such similarity as

$$||\boldsymbol{\varepsilon}_{ij}||^2 \leqslant \gamma^2 \quad \text{where} \quad \boldsymbol{\varepsilon}_{ij} = \mathbf{z}_j - \mathbf{z}_i. \tag{5}$$

In [1], γ is chosen as a small threshold dependent on the number of pixels (n) in each patch. The relationship between similar patches shown in (5) is based on the underlying noise-free patches that are not known in practice. In [2], this expression was extended to define photometric similarity between the corresponding *noisy* patches $\mathbf{y}_j$ as

$$||\widetilde{\boldsymbol{\varepsilon}}_{ij}||^2 \leqslant \gamma_n^2 = \gamma^2 + 2\sigma^2 n \quad \text{where} \quad \widetilde{\boldsymbol{\varepsilon}}_{ij} = \mathbf{y}_j - \mathbf{y}_i. \tag{6}$$

The N_i values can be then directly estimated from the noisy image as the number of $\mathbf{y}_j$ patches (including $\mathbf{y}_i$) that satisfy the aforementioned criterion. Note that the condition for photometric similarity, as defined here, is stricter than that for geometric similarity. As such, photometric similarity can be expected to imply geometrically similar as well.

The bounds expression in (3) thus takes into account the complexity of the image patches present in the image, as well as the redundancy level and the noise variance corrupting the image. In [1], the bound was shown to characterize the performance of the optimal affine-biased denoising method. In particular, for the WGN, the right-hand side of (3) is the performance achieved by the optimal linear minimum mean-squared-error (LMMSE) estimator, with $\mathbf{J}_i$ and $\mathbf{C}_{\mathbf{z}}$ being the parameters of the estimator. The Wiener filter is, in fact, the LMMSE estimator that achieves this lower bound [29]. Thus, a patch-based Wiener filter, where the parameters are accurately estimated, can lead to near-optimal denoising. This forms the basis of our approach. We outline the theory behind the proposed approach next.

B. Derivation and Analysis

Irrespective of the noise characteristics, the expression in (3) leads to the lowest MSE theoretically achievable by any patch-based denoising method. This expression was derived in [1] assuming that the underlying unknown image patches $\mathbf{z}_i$ are (independent) realizations of a random variable $\mathbf{z}$. Furthermore, image patches that are *geometrically* similar were considered to be sampled from the same (unknown) probability density function (pdf) $p(\mathbf{z})$. Thus, we need to first identify geometrically similar patches within the image and group them. As such, image patches within each cluster will exhibit similar structure, although the actual intensity values can be quite different, as shown in Fig. 2. For now, we assume that such grouping is made available through some oracle clustering method.[1] When the corrupting noise is the WGN, the LMMSE estimate of each patch $\mathbf{z}_i$ from its noisy observation $\mathbf{y}_i$ has the following form [29]:

$$\widehat{\mathbf{z}}_i = \bar{\mathbf{z}} + \mathbf{C}_{\mathbf{z}}\mathbf{C}_{\mathbf{y}}^{-1}(\mathbf{y}_i - \bar{\mathbf{z}}) \tag{7}$$

where $\bar{\mathbf{z}}$ and $\mathbf{C}_{\mathbf{z}}$ are the first and second moments of the pdf $p(\mathbf{z})$ from which all patches geometrically similar to $\mathbf{z}_i$ are assumed to be independently sampled. The covariance of the (geometrically similar) noisy image patches can be expressed as

$$\mathbf{C}_{\mathbf{y}} = \mathbf{C}_{\mathbf{z}} + \sigma^2\mathbf{I}. \tag{8}$$

Thus, the parameters of the LMMSE filter remain the same for all patches that are considered to be similar in structure. Such an estimator is unbiased, i.e., $E[\hat{\mathbf{z}}_i] = \bar{\mathbf{z}}$, and the estimation error can be expressed as

$$E\left[||\mathbf{z}_i - \widehat{\mathbf{z}}_i||^2\right] = \mathrm{Tr}\left[\left(\mathbf{C}_{\mathbf{z}}^{-1} + \frac{\mathbf{I}}{\sigma^2}\right)^{-1}\right]. \tag{9}$$

A similar approach was applied to the problem of super-resolution and demosaicing in [30]. The aforementioned estimator achieves the bounds in (3) when $N_i = 1$, which is the case when *photometric* similarities are not observed in the input image. However, in general, natural images exhibit some level of photometric redundancy. Exploiting such repetitions forms the core of many denoising methods [8]–[12], [16], where photometrically similar patches are considering to be multiple observations of a single latent patch with the differences arising (ideally) due to noise only. Most similarity-based methods thus identify *photometrically similar* patches within the noisy image to perform denoising, with the most similar patches exerting the greatest influence in the denoising process. Our denoising framework can be also generalized to exploit such *photometric* redundancies within any given noisy image. Moreover, these patches need not to necessarily be spatially proximal (as in [8] and [9]), thereby giving rise to a so-called *nonlocal* patch-based Wiener filter for denoising.

As mentioned earlier, photometric similarity among patches, as required to exploit redundancy, is a stricter condition than the geometric similarity property used for clustering. We therefore require an additional step of identifying the $\mathbf{y}_j$ patches that are

[1]We specify a practical method in Section III-A and in [1] and [15].

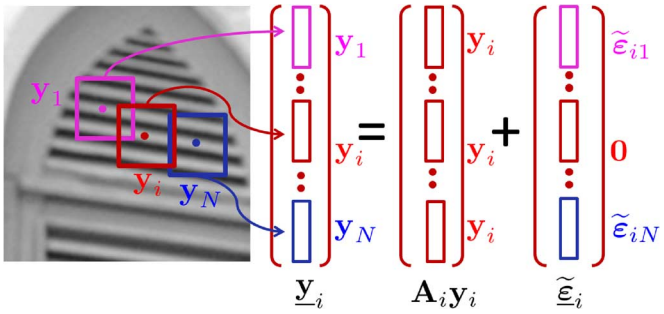


Fig. 3. Illustration of the data model formed by collecting all photometrically similar patches. Here, $\mathbf{y}_i$ is the reference patch, and $\mathbf{y}_1, \ldots, \mathbf{y}_N$ are patches that satisfy the similarity condition of (6). All these patches are then accumulated to form the data model of (10).

photometrically similar to any given patch $\mathbf{y}_i$. These $\mathbf{y}_j$ patches all satisfy the condition of (6) and can be expressed as (see Fig. 3)

$$\begin{aligned}\underline{\mathbf{y}}_i &= \mathbf{A}_i\mathbf{y}_i + \underline{\widetilde{\boldsymbol{\varepsilon}}}_i = \mathbf{A}_i(\mathbf{z}_i + \boldsymbol{\eta}_i) + (\underline{\boldsymbol{\varepsilon}}_i + \underline{\boldsymbol{\eta}}_i - \mathbf{A}_i\boldsymbol{\eta}_i) \\ &= \mathbf{A}_i\mathbf{z}_i + \underbrace{\underline{\boldsymbol{\varepsilon}}_i + \underline{\boldsymbol{\eta}}_i}_{\underline{\boldsymbol{\zeta}}_i} \end{aligned} \tag{10}$$

where $\underline{\mathbf{y}}_i \in \Re^{nN_i}$ is a vector formed by concatenating all (e.g., N_i) $\mathbf{y}_j$ patches that are photometrically similar to $\mathbf{y}_i$, $\underline{\boldsymbol{\eta}}_i \in \Re^{nN_i}$ is the corresponding noise patches stacked together, $\underline{\boldsymbol{\varepsilon}}_i$ and $\underline{\widetilde{\boldsymbol{\varepsilon}}}_i$ are nN_i-sized vectors consisting of concatenated difference vectors $\boldsymbol{\varepsilon}_{ij}$ and $\widetilde{\boldsymbol{\varepsilon}}_{ij}$ of (5) and (6), respectively, and $\mathbf{A}_i = [\mathbf{I}, \ldots, \mathbf{I}]^T \in \Re^{nN_i \times nN_i}$ is the matrix formed by vertically stacking N_i identity matrices, each of size $n \times n$. Letting $\mathbf{C}_{\underline{\boldsymbol{\zeta}}_i} \in \Re^{nN_i \times nN_i}$ denote the covariance matrix for the error vector $\underline{\boldsymbol{\zeta}}_i = \underline{\boldsymbol{\varepsilon}}_i + \underline{\boldsymbol{\eta}}_i$, we can write the corresponding LMMSE filter as [29]

$$\begin{aligned}\widehat{\mathbf{z}}_i &= \bar{\mathbf{z}} + \mathbf{C}_{\mathbf{z}}\mathbf{A}_i^T\left(\mathbf{A}\mathbf{C}_{\mathbf{z}}\mathbf{A}_i^T + \mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}\right)^{-1}(\underline{\mathbf{y}}_i - \mathbf{A}_i\bar{\mathbf{z}}) \\ &= \bar{\mathbf{z}} + \left(\mathbf{C}_{\mathbf{z}}^{-1} + \mathbf{A}_i^T\mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}^{-1}\mathbf{A}_i\right)^{-1}\mathbf{A}_i^T\mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}^{-1}(\underline{\mathbf{y}}_i - \mathbf{A}_i\bar{\mathbf{z}}). \end{aligned} \tag{11}$$

As before, parameters $\bar{\mathbf{z}}$ and $\mathbf{C}_{\mathbf{z}}$ are the moments obtained from the geometrically similar patches within each cluster. The aforementioned expression leads to the optimal estimator (in terms of the MSE) when the error vector $\underline{\boldsymbol{\zeta}}_i$ can be assumed to be normal distributed. Under sufficiently strong WGN, such an assumption is justified as the $\boldsymbol{\varepsilon}_{ij}$ vectors are small by definition [see (5)] in comparison with the noise.

The expression in (11) can be further simplified if we assume the components of $\boldsymbol{\varepsilon}_{ij}$ vectors to be i.i.d.. Along with the fact that the $\boldsymbol{\varepsilon}_{ij}$ vectors are independent of the $\boldsymbol{\eta}_i$ noise vectors, this results in a diagonal $\mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}$ (derivation in Appendix A), i.e.,

$$\mathbf{C}_{\underline{\boldsymbol{\zeta}}_i} = \mathbf{C}_{\underline{\boldsymbol{\varepsilon}}_i} + \mathbf{C}_{\underline{\boldsymbol{\eta}}_i} = \begin{bmatrix} \ddots & & \mathbf{0} \\ & \delta_{ij}^2\mathbf{I} & \\ \mathbf{0} & & \ddots \end{bmatrix} \tag{12}$$

where

$$\delta_{ij}^2 = \frac{1}{n}E\left[\|\mathbf{y}_i - \mathbf{y}_j\|^2\right] - \sigma^2. \tag{13}$$

Denoting $w_{ij} = \delta_{ij}^{-2}$, the LMMSE estimator of (11) can be alternately expressed as (see Appendix B)

$$\widehat{\mathbf{z}}_i = \bar{\mathbf{z}} + \left(\mathbf{C}_{\mathbf{z}}^{-1} + \sum_{j=1}^{N_i} w_{ij}\mathbf{I}\right)^{-1}\sum_{j=1}^{N_i} w_{ij}(\mathbf{y}_j - \bar{\mathbf{z}}) \tag{14}$$

$$= \bar{\mathbf{z}} + \left(\frac{\mathbf{C}_{\mathbf{z}}^{-1}}{\sum_{j=1}^{N_i} w_{ij}} + \mathbf{I}\right)^{-1}\sum_{j=1}^{N_i}\frac{w_{ij}}{\sum_{j=1}^{N_i} w_{ij}}(\mathbf{y}_j - \bar{\mathbf{z}}). \tag{15}$$

It is shown in (14) that a weighted contribution of each similar patch is used to come up with a denoised estimate for each $\mathbf{z}_i$ where the contributing factor of any $\mathbf{y}_j$ gets larger with increasing similarity (δ_{ij}^{-2}). The error covariance matrix for such an estimator is approximately [29]

$$\mathbf{C}_e \approx \left(\mathbf{C}_{\mathbf{z}}^{-1} + \sum_{j=1}^{N_i} w_{ij}\mathbf{I}\right)^{-1}. \tag{16}$$

Comparing the aforementioned expression to the bounds formulation of (3), we can see that, when $\sum_j w_{ij} = N_i/\sigma^2$, the estimator achieves the bound.[2] Such a scenario arises when the underlying noise-free image contains N_i exact replicas for patch $\mathbf{z}_i$, i.e., when $E[\|\mathbf{z}_i - \mathbf{z}_j\|^2] = 0$. In practice, such levels of redundancy can be rare, and even if very similar patches exist, identifying such patches can be challenging under noise contamination, resulting in higher MSE.

Although (15) provides a nice formulation for the estimator, it can lead to mathematical instabilities as the covariance matrix $\mathbf{C}_{\mathbf{z}}$ can be rank deficient or ill conditioned. To circumvent the possibility of errors due to inversion, we make use of the matrix inversion lemma [31] to state an alternate form of the LMMSE estimator (see Appendix B for entire derivation), i.e.,

$$\widehat{\mathbf{z}}_i = \bar{\mathbf{z}} + \left[\mathbf{I} - \left(\sum_j w_{ij}\mathbf{C}_{\mathbf{z}} + \mathbf{I}\right)^{-1}\right]\sum_j\frac{w_{ij}}{\sum_j w_{ij}}(\mathbf{y}_j - \bar{\mathbf{z}})$$

$$= \left[\sum_{j=1}^{N_i}\frac{w_{ij}\mathbf{y}_j}{\sum_{j=1}^{N_i} w_{ij}}\right] \tag{17}$$

$$+ \left[\sum_{j=1}^{N_i}\frac{w_{ij}}{\sum_{j=1}^{N_i} w_{ij}}\left(\sum_{j=1}^{N_i} w_{ij}\mathbf{C}_{\mathbf{z}} + \mathbf{I}\right)^{-1}(\bar{\mathbf{z}} - \mathbf{y}_j)\right]. \tag{18}$$

This leads to an interesting formulation where the first part of (18) is exactly the expression for the NLM [8], [9] filter, although the NLM was not the basis for our derivation. Since $\bar{\mathbf{z}}$ is obtained from all *geometrically* similar patches in a cluster, it can be considered as a naive denoised estimate, which is oversmoothened. The latter part of the expression in (18) filters the residuals between the noisy similar patches and this naive estimate. These filtered residuals are then added to the weighted mean of photometrically similar patches. The latter term can

[2]For the sake of clarity, we assume here that the mean and the covariance in each cluster are known and, hence, the estimator remains unbiased. In practice, these parameters are estimated from a limited number of noisy patches resulting in higher MSE than that predicted by the lower bound [see (16) or (3)].

be then thought of as a correction that improves the estimate by a directional filtering of the residuals based on their shared geometric structure. This suppresses the noise further, while restoring more of the finer details in the image patches. When the structural information of image patches are ignored (i.e., all structures are equally probable, implying a large determinant of $\mathbf{C_z}$), we obtain the NLM filter as a suboptimal approximation (in terms of the MSE) of our formulation in (18). Note that we express patch-based locally optimal Wiener (PLOW) filter as a summation of two related parts (NLM and residual filtering) only to gain theoretical insights. In practice, we use (17) to obtain the denoised estimate for each patch.

Until now, we have presented and analyzed the theoretical basis for our proposed approach. In the next section, we provide a practical outline for our algorithm that details the estimation of each parameter of the proposed filter from a given noisy image.

Algorithm 1 PLOW denoising

Input: Noisy image: $\mathbf{Y}$

Output: Denoised image: $\widehat{\mathbf{Z}}$

1: Set parameters: patch size $n = 11 \times 11$, number of clusters $K = 15$;
2: Estimate noise standard deviation $\hat{\sigma}$ [see (21)];
3: Set parameter: $h^2 = 1.75\hat{\sigma}^2 n$;
4: $\mathbf{Y}^0 \Leftarrow$ Prefilter image to obtain pilot estimate;
5: $\{\mathbf{y}_i, \mathbf{y}_i^0\} \Leftarrow$ extract overlapping patches of size n from $\mathbf{Y}$ & $\mathbf{Y}^0$;
6: $\mathbf{L} \Leftarrow$ compute LARK features for each $\mathbf{y}_i^0$;
7: $\mathbf{\Omega}_k \Leftarrow$ geometric clustering with K-means $(\mathbf{L}, K)$;
8: **foreach** *Cluster* $\mathbf{\Omega}_k$ **do**
9: Estimate mean patch $\widehat{\bar{\mathbf{z}}}$ from $\mathbf{y}_i^0 \in \mathbf{\Omega}_k$ [see (19)];
10: Estimate cluster covariance $\widehat{\mathbf{C}}_\mathbf{z}$ from $\mathbf{y}_i^0 \in \mathbf{\Omega}_k$ [see (20)];
11: **foreach** *Patch* $\mathbf{y}_i^0 \in \mathbf{\Omega}_k$ **do**
12: $\mathbf{y}_j^0 \Leftarrow$ identify photometrically similar patches [see (6)];
13: $w_{ij} \Leftarrow$ compute weights for all $\mathbf{y}_j^0$ [see (23)];
14: $\widehat{\mathbf{z}}_i \Leftarrow$ estimate denoised patch using $\mathbf{y}_j$ [see (17)];
15: $\mathbf{C}_{e_i} \Leftarrow$ calculate estimate error covariance [see (16)];
16: **end**
17: **end**
18: $\widehat{\mathbf{Z}} \Leftarrow$ aggregate multiple estimates from all $\{\widehat{\mathbf{z}}_i\}$ and $\{\mathbf{C}_{e_i}\}$ [see (27)].

III. Parameter Estimation for Denoising

Our proposed denoising framework, graphically outlined in Fig. 1, requires us to infer various parameters from the observed noisy image. The procedure is algorithmically represented in Algorithm 1. We first identify geometrically similar patches within the noisy image. Once such patches are identified, we can use these patches to estimate the moments ($\bar{\mathbf{z}}$ and $\mathbf{C_z}$) of the cluster, taking care to account for noise (steps 9 and 10 of Algorithm 1). Next, we identify the photometrically similar patches and calculate weights w_{ij} that control the amount of influence that any given patch exerts on denoising patches similar to it. These parameters are then used in (18) to denoise each patch. Since we use overlapping patches, multiple estimates are obtained for pixels lying in the overlapping regions. These multiple estimates are then optimally aggregated to obtain the final denoised image. Below, we describe each step in greater detail.

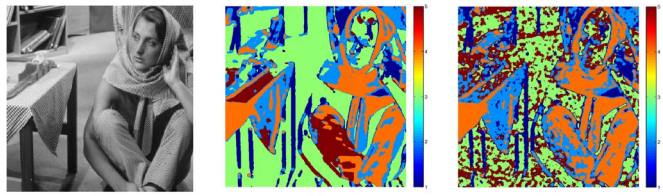


Fig. 4. Clustering based on the geometric similarity of patches illustrated on noise-free and noisy Barbara images with a standard deviation of 15. Note how clustering in the noisy image largely corresponds to that from the noise-free image. (a) Barbara image. (b) Noise-free clusters. (c) Noisy clusters.

A. Geometric Clustering

In Section II, our proposed filter was derived assuming geometrically similar patches to be sampled from some unknown pdf. So far, we have assumed such clustering to be available to us from an "oracle." To perform practical clustering, we need to identify features that capture the underlying geometric structure of each patch from its noisy observations. Such features need to be robust to the presence of noise, as well as to differences in contrast and intensity among patches exhibiting similar structural characteristics. An example of such variations among geometrically similar patches is shown in Fig. 2. Possible choices of features include contrast-adjusted image patches [30] or principal components in conjunction with predetermined clustering guides [32]. For our purposes, where the image patches can be considerably noisy, we make use of the locally adaptive regression kernels (LARKs) introduced for denoising in [7] and subsequently adapted as features for geometric clustering [15] and object detection [33]. We refer the interested reader to [7], where the design of the kernels is covered in detail.

Using the LARK features, we run K-means [34] to cluster the noisy image into regions containing *geometrically similar* patches. In Fig. 4, we illustrate the robustness of clustering using LARK features, [3] even in the presence of noise with a standard deviation of 15. Note that the five clusters from the noisy Barbara image are largely in keeping with those obtained from the noise-free image.

As noted in [15], the number of clusters chosen affects the denoising result. In general, too few clusters can lead to structurally dissimilar patches being clustered together resulting in the incorrect estimation of the moments. On the other hand, too many clusters lead to too few patches within each cluster, making the moment estimation process less robust. Fortunately, the denoised output is not too sensitive to the choice of the number of clusters (K). In our experiments, we found that using a fixed value of $K = 15$ yields good results for any given image, with the MSE fairly close to that obtained by tuning the number of clusters for that particular image.

[3]In Fig. 4, we color code the center pixel y_i of each patch $\mathbf{y}_i$ depending on cluster $\mathbf{\Omega}_k$ in which the corresponding LARK feature lies. Each patch therefore belongs to a single cluster, which we denote as $\mathbf{y}_i \in \mathbf{\Omega}_k$.

B. Estimating Cluster Moments

Once the image is segmented into structurally similar regions, we estimate the moments, namely, mean and covariance, from the noisy member patches of each cluster. Since the $\boldsymbol{\eta}_i$ noise patches are assumed to be zero mean i.i.d., the mean of the underlying noise-free image can be approximated by the expectation of the noisy patches within each cluster as

$$\hat{\bar{\mathbf{z}}} = E[\mathbf{y}_i \in \boldsymbol{\Omega}_k] \approx \frac{1}{M_k} \sum_{\mathbf{y}_i \in \boldsymbol{\Omega}_k} \mathbf{y}_i \tag{19}$$

where $\boldsymbol{\Omega}_k$ denotes the kth cluster with cardinality M_k. Note that the stability of this estimate is dependent on M_k. If too few patches are present, the mean vector will remain noisy.

The covariance matrix $\mathbf{C_z}$ is also estimated from the noisy patches within the cluster. For this, we make use of the relation between the covariance of the noisy $(\mathbf{C_y})$ and noise-free patches $(\mathbf{C_z})$ from (8). We thus need to first estimate $\mathbf{C_y}$. Covariance estimation is an active research area with a wide variety of applications [35]–[37]. The simplest of such estimators, i.e., the sample covariance, is the maximum likelihood estimate. Although other estimators, for example, bootstrapping [35] and shrinkage-based [36] methods, exist, we found no discernible improvement in the denoising performance when more complex estimators were used. When the number of samples (patches in a cluster) are few compared with the dimensionality (number of pixels in each patch), the sample covariance can be unstable. For such cases, robust estimators proposed in [37] may also be used.

Working with the sample covariance $\widehat{\mathbf{C}}_\mathbf{y}$, we estimate the covariance of the underlying noise-free patches as

$$\widehat{\mathbf{C}}_\mathbf{z} = \left[\widehat{\mathbf{C}}_\mathbf{y} - \sigma^2 \mathbf{I}\right]_+ \tag{20}$$

where σ^2 is the noise covariance and $[\mathbf{X}]_+$ denotes matrix $\mathbf{X}$ with its negative eigenvalues replaced by zero (or a very small positive value), as done in [2]. For this, we need to accurately estimate the noise standard deviation first. Here, we use a gradient-based estimator as [38]

$$\hat{\sigma} = 1.4826 \,\text{median}\left(|\nabla\mathbf{Y} - \text{median}(\nabla\mathbf{Y})|\right) \tag{21}$$

where $\nabla\mathbf{Y}$ is the vectorized form of the gradient of the input image $\mathbf{Y}$. The gradient image $\nabla\mathbf{Y}$ is calculated as

$$\nabla\mathbf{Y} = \frac{1}{\sqrt{6}} \text{vec}\left(\mathbf{Y} \star \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}\right). \tag{22}$$

Here, $\text{vec}(.)$ denotes the vectorization operation (column- or rowwise) and the convolution $(\star)$ operation simply implies the addition of the forward gradients in the horizontal and vertical directions. In [2], we showed that the shrinkage estimator of (20) is accurate enough to compute the bounds directly from the noisy image. In the present case of denoising, a similar observation holds.

We point out that, in deriving bounds [1] (3) and our PLOW filter (see Section II-A), we assume that the underlying noise-free patches $\mathbf{z}_i$ are independent samples of a random variable $\mathbf{z}$. In practice, when working with overlapping patches, this assumption is not strictly accurate. As with all other patch-based methods, since we estimate each patch independently without explicitly taking into account information from *estimates* of other overlapping patches, the estimation framework is in line with the assumption of the independence of underlying (noise-free) patches. However, in estimating the covariance matrix (in PLOW and also for the bounds in [1] and [2]), we do not enforce independence on the patches, and the covariance matrix estimated from overlapping patches is not necessarily diagonal. Therefore, both in our bounds and our current paper, the correlation among the underlying noise-free patches are implicitly taken into account.

The issue is more subtle when considering the overlapping noise patches. In that case, our assumption of $\boldsymbol{\eta}_i$ patches being independent is a simplification mainly used for the ease of mathematical derivation. By now, this is a standard practice employed in all patch-based methods (NLM [8], BM3D [12], NLSM [16], etc.), where information shared among overlapping noisy patches are not exploited. We hazard a guess that one can expect some modest improvement in the performance if such information are accounted for in the denoising process. Doing so is however nontrivial and may be a good direction for future research.

The parameters estimated from each cluster of the image can be directly used for denoising using (7). However, as mentioned earlier, the performance can be improved by exploiting the self-similarities within an image. Using photometrically similar patches can then contribute to the better denoising of the reference patch. We describe this process next.

C. Calculating Weights for Similar Patches

In our paper, we first identify patches within the noisy image that are *photometrically similar* to a given reference patch. Once the similar patches are identified for a given reference patch, we perform denoising with the more similar patches exerting greater influence in the denoising process. This is ensured by the analytically derived weight w_{ij}, which determines the contributing factor for patch $\mathbf{y}_j$ in denoising the reference patch $\mathbf{y}_i$. Weight w_{ij} is related to the inverse of the expected squared ℓ_2 distance between the underlying noise-free patches and a noise term [see (13)].

Although the weight calculation in (13) is statistically well motivated, it is impractical as we need to approximate $E[\|\mathbf{y}_i - \mathbf{y}_j\|^2]$ from a single $\mathbf{y}_i$-and- $\mathbf{y}_j$ pair. Here, we approximate this similarity measure (see Appendix C for derivation) by [4]

$$w_{ij} \approx \frac{1}{\sigma^2} \exp\left\{-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{h^2}\right\} \tag{23}$$

where the scalar multiplier $1/\sigma^2$ also ensures that the denoiser defaults to that of (7) when no photometrically similar patches are observed, i.e., when $\mathbf{y}_j = \mathbf{y}_i$. The smoothing parameter h^2 is a positive parameter that controls the rate at which the contributing factor is driven to zero as the patches become less similar. Although tunable in general, in our algorithm, this parameter is kept fixed at $h^2 = 1.75\sigma^2 n$. This was empirically found

[4]Note that the expression in (23) is similar to that introduced in [8]. In Appendix C, we motivate this formulation statistically and derive it as an approximation to the distance metric in (13).

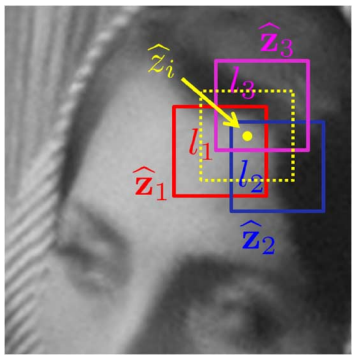


Fig. 5. Illustration of how a pixel is estimated multiple times due to overlapping patches. Here, we show three such overlapping patches. In each estimated patch $\widehat{\mathbf{z}}_r$ (here $r = 1, 2, 3$), the same pixel is estimated as its l_rth pixel, which we denote as $\hat{z}_{rl}$. These estimates are combined to form the final estimate $\hat{z}_i$.

to be close to the optimal h^2 value for a wide range of images and across different noise levels.

Note that photometrically similar patches are necessarily geometrically similar as well, and hence, we could limit our search within the cluster of the reference patch. However, errors in clustering (see Fig. 4) can limit the number of similar patches identified. On the other hand, scanning the entire image can be time consuming. Consequently, we restrict ourselves to a relatively small search window (30×30 pixels). Apart from speed considerations, as the motivation was in [8], this also leads to better performance [39].

D. Aggregating Multiple Pixel Estimates

Until now, we have estimated all the parameters needed to perform the filtering of (18). The filter is run on a per-patch basis (although parameters are estimated from multiple patches), yielding denoised estimates for each patch of the noisy input. To avoid block artifacts at the patch boundaries, the patches are chosen to overlap each other. As a result, we obtain multiple estimates for the pixels lying in the overlapping regions. This is shown in Fig. 5 where z_i is estimated multiple times as a part of different patches. These multiple estimates need to be aggregated to form a final denoised image.

The simplest method of aggregating such multiple estimates is to average them. However, such naive averaging will lead to an oversmoothened image. Alternatively, in keeping with earlier formulation, we can combine the multiple estimates in an LMMSE scheme that takes into account the relative confidence in each estimate. The covariance of our proposed estimator is approximated by [see (16)]

$$\mathbf{C}_e \approx \left(\widehat{\mathbf{C}}_{\mathbf{z}}^{-1} + \sum_{j=1}^{N_i} w_{ij} \mathbf{I} \right)^{-1}. \tag{24}$$

Let us denote $\hat{z}_{rl}$ as the denoised estimate for the lth pixel in the rth patch (see Fig. 5). Then, variance v_{rl} associated with the lth pixel estimate is given by the lth diagonal element of $\mathbf{C}_e$. Concatenating the multiple (i.e., R) estimates $\hat{z}_{rl}$ in vector $\widehat{\mathbf{z}}_{ir}$, we can write

$$\widehat{\mathbf{z}}_{ir} = \mathbf{1} z_i + \boldsymbol{\tau}_{ir} \tag{25}$$

where $\mathbf{1} \in \Re^R$ is a vector of all ones and $\boldsymbol{\tau} \in \Re^R$ is the error vector assumed to be zero mean Gaussian with covariance $\mathbf{C}_{\boldsymbol{\tau}} = \text{diag}[\ldots\ v_{rl}\ \ldots]$. The LMMSE estimate for the ith pixel of the image is then

$$\begin{aligned} \hat{z}_i &= \left(\sigma_z^{-2} + \mathbf{1}^T \mathbf{C}_{\boldsymbol{\tau}}^{-1} \mathbf{1}\right)^{-1} \mathbf{1}^T \mathbf{C}_{\boldsymbol{\tau}}^{-1} \widehat{\mathbf{z}}_{ir} \\ &= \frac{\sum_{r=1}^{R} v_{rl}^{-1} \hat{z}_{rl}}{\sum_r v_{rl}^{-1} + \sigma_z^{-2}} \end{aligned} \tag{26}$$

where σ_z^2 is the variance of z_i, which forms the prior information. Note that, although we estimate the covariance of the image patches ($\widehat{\mathbf{C}}_{\mathbf{z}}$), this does not provide us with a pixelwise variance estimate σ_z^2. This is a result of considering overlapping patches where any given pixel z_i can lie in *different locations in different patches* (see Fig. 5). Moreover, the overlapping patches may also lie in different clusters where the $\widehat{\mathbf{C}}_{\mathbf{z}}$ matrices can also be different. In the absence of a particular σ_z, we consider all possible z_i values (within the intensity range [0–255]) to be equally likely, leading to the variance of the discrete uniform distribution $\sigma_z^2 = (256^2 - 1)/12 \Rightarrow \sigma_z^{-2} = 0$. This reduces (26) to the weighted least-square solution, i.e.,

$$\hat{z}_i = \sum_{r=1}^{R} \frac{v_{rl}^{-1} \hat{z}_{rl}}{\sum_r v_{rl}^{-1}} \tag{27}$$

where the number of estimates (R) of the ith pixel depends on the size of the patch (n), the amount of overlap,[5] and the position of the pixel in the patches (pixels toward the edge of a patch are more likely to lie in overlapping regions). Here, r indexes only those R patches that include the ith pixel of the image and position l of the ith pixel is dependent upon the patch r being considered. Such a weighted least-square estimate obtained using the error variance from multiple estimates (v_{rl}), dating at least back to [40], also forms the basis for aggregation in [12], [41], and [42]. In every case, such aggregation leads to an improved final pixel estimate compared with other naive approaches (e.g., simple averaging).

As shown, the necessary parameters of our proposed filter can be estimated from a given noisy image. The accuracy of estimating such parameters is dependent on the strength of the noise corrupting the image. Noise affects different parameter estimation steps differently. The moment estimation step is dependent on the ability of the clustering step to classify structurally similar patches. Although the LARK features are quite robust, errors in clustering due to noise cannot be fully avoided. This is directly shown in Fig. 4, where differences in clustering the noisy and noise-free images are apparent.

Furthermore, even with accurate clustering, noise causes the eigenvalues of the sample covariance matrix $\mathbf{C}_{\mathbf{y}}$ to be unequally shifted (some are shifted by more than σ^2, whereas others are shifted less) by a factor dictated by the size of the patches, the number of patches within a cluster, and the strength of the noise

[5]Note that a larger overlap implies more patches for clustering, moment estimation, and higher levels of redundancy among image patches. This makes the estimation process robust and allows for improved performance. However, this performance comes at the cost of speed. A reasonable approach is to use all patches (at 1-pixel shifts) for parameter estimation and denoise only every few (overlapping) patches. The aggregation step can be then used to reconstruct the entire image. The performance of such a scheme, visually and MSE-wise, is reasonably close to that obtained by denoising densely.

[43]. With strong noise, the lower eigenvalues of the sample covariance are considerably underestimated, leading to a lower rank estimate of $\mathbf{C_z}$ in (20). Using such a $\widehat{\mathbf{C}}_\mathbf{z}$ in our denoising framework results in the loss of finer details as the second term in (18) is projected onto a lower dimensional subspace.

Additionally, the weight calculation process of (23) is quite sensitive to noise. Identifying photometrically similar patches becomes challenging in the presence of strong noise [2], [44], which, in turn, influences the similarity measure calculation of (23). To alleviate these detrimental effects of noise, we prefilter the image once before the parameters of the proposed framework are learned. Note that prefiltering to obtain a "pilot" estimate is quite typical of competing approaches [12], [19] and is necessary only for strong noise. While any denoising method can be employed, for consistency, as a preprocessor, we apply our algorithm with a reduced noise variance estimate (heuristically chosen as $0.75\hat{\sigma}^2$) to ensure that finer details are not lost in the prefiltered image. The necessary filter parameters are then learned from the resultant noise-suppressed image. These parameters are then applied to the original noisy image for denoising. For strong noise, such prefiltering invariably results in noticeably improved denoising performance. The performance of our method is demonstrated next.

IV. Results

Here, we evaluate the proposed denoising method through experiments on various images at different noise levels. Since our method is motivated by our bounds formulation [1], we first compare the ideal denoising performance of our method (using "oracle" parameters) with the MSE predicted by the bounds. Later, we estimate the parameters directly from the noisy images, as outlined in Section III, and compare those results with various popular denoising methods. We also apply our method, with a minor modification, to color images. Finally, we address the practical case of denoising real noisy images where the noise characteristics are unknown and not necessarily Gaussian, or uncorrelated. In each case, we will show that our results are comparable, in terms of the MSE [peak signal-to-noise ratio (PSNR[6])], SSIM [45], and the recently introduced no-reference quality metric Q [46] (wherever applicable), with those obtained by state-of-the-art denoising methods and are, in many cases, visually superior.

Since our method was specifically designed with the aim of achieving the theoretical limits of the performance, we first compare our results to the predicted performance bounds [1]. For this first experiment, we compute the "oracle" denoising parameters from the noise-free images. To be precise, we compute the structure-capturing LARK features from the noise-free image and perform clustering. These "oracle" clusters are then used to estimate moments $\bar{\mathbf{z}}$ and $\mathbf{C_z}$ from the latent image. We also use the ground-truth image to identify the photometrically similar patches and compute weights w_{ij} for each noise-free reference patch. The final denoising using the "oracle" parameters is, of course, applied to the noisy image.

[6]PSNR is measured in decibels and is calculated as $10\log_{10}(255^2/\text{MSE})$ for images with intensity range of [0–255]. An improvement of 1 dB reduces the MSE by approximately 20%.

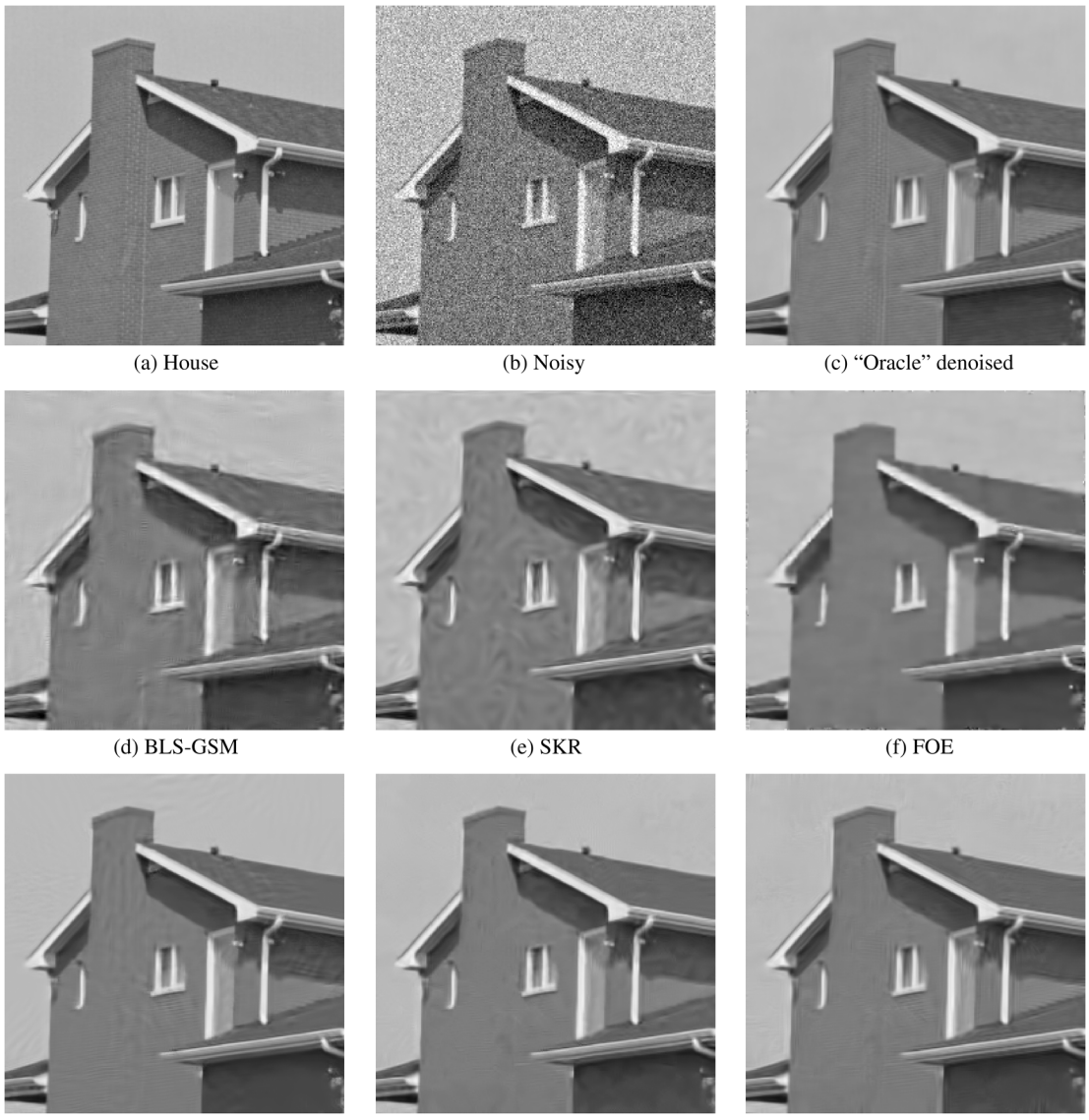
Not surprisingly, the results are quite impressive in terms of denoising achieved with finer details being retained at the same time. The MSE obtained for the Barbara and house images are 49.42 and 27.97, respectively. For the Barbara image, the lowest MSE predicted by the denoising bound (MSE of 50.24) is achieved,[7] whereas for the house image, the bound (MSE of 14.82) is still lower. One reason for such discrepancy between the theoretical prediction and what we obtain in practice, even with oracle parameters, is that the theory in the bounds [1] is built on the assumption of *exact* replicas of patches being observed. However, in reality, such replicas are rare, even in "noise-free" images.[8] It is encouraging to note that the optimal MSE is well below the state of the art for the house image for which the bounds predict the possibility of improved performance [1]. On the other hand, the optimal performance for the more complicated Barbara image is comparable with that of BM3D, in keeping with the bounds predicting little improvement to be gained. The improvement is also visually apparent at the finely detailed regions where parameter estimation is more error prone [see Fig. 6(c)].

Having established that our method performs near-optimal denoising with "oracle" knowledge of parameters, we experiment with the more practical case when the parameters are directly estimated from the noisy image, as outlined in Section III. In Figs. 6–8, we compare our results to various high-fidelity methods for image denoising. In Table I, we quantify the performances for a variety of benchmark images, across different noise levels, with different performance measures (PSNR, SSIM [45], and the no-reference quality metric Q [46]). The best results are shown in bold for ease of comparison. Additionally, we also highlight results where the difference from the best results are statistically insignificant (0.1 dB in PSNR and 0.01 for SSIM). In terms of all the quantitative measures, Table I shows that our method is quite comparable with BM3D [12] and NLSM [16]. While BM3D's high performance has not been well justified on theoretical grounds yet, the steps involved in NLSM can be quite complex and time consuming (about 170 s on a 256×256 grayscale image). Our method is however well motivated and provides a statistical explanation for its performance.

In terms of visual quality, our method is comparable with NLSM and BM3D, even outperforming them in many cases where images exhibit higher levels of redundancy. This can be observed in Fig. 7, where our result is more visually pleasing when compared with NLSM and BM3D, both of which produce more structured artifacts in the smoother floor and face regions (better noticed when viewed at native resolutions online at http://users.soe.ucsc.edu/~priyam/PLOW/). Images containing more semistochastic textures typically exhibit lower levels of patch redundancy. For such images, BM3D typically does a better job of denoising. However, even in such cases, our denoising results are visually comparable with the state of the art.

[7]It may seem here that the lower bounds are breached, albeit marginally, for the Barbara image. However, in [1], the bounds were calculated with five clusters, whereas we use 15 clusters here. It was shown in [1] that using more clusters reduces the bounds further, although the reduction is nominal.

[8]The term "noise-free" here is an idealization used to imply the original image before noise is added. In general, images captured are invariably noisy due to the imaging process [3]. That images considered to be ground-truth also contain noise, albeit in low strengths, has been illustrated in [47].



(a) House (b) Noisy (c) "Oracle" denoised

(d) BLS-GSM (e) SKR (f) FOE

(g) NLSM (h) BM3D (i) PLOW

Fig. 6. Comparison of denoising results on noisy house image corrupted by WGN of $\sigma = 25$. (a) Original image, (b) noisy input, (c) proposed method with "oracle" parameters (MSE of 27.97), (d) BLS-GSM [23] (MSE of 49.16), (e) SKR [7] (MSE of 47.69), (f) FOE [20] (MSE of 50.36), (g) NLSM [16] (MSE of 32.31), (h) BM3D [12] (MSE of 33.36), and (i) PLOW (MSE of 34.51). High-resolution images can be viewed at http://users.soe.ucsc.edu/~priyam/PLOW/.

As a next step, we apply our method to the problem of denoising color images. In [48], it was pointed out that frequencies at which the human eye perceives each of the red, green, and blue (RGB) colors have considerable overlap. Consequently, many color denoising methods take into account such dependencies, either implicitly or explicitly. Mairal *et al.* [49] illustrated the usefulness of enforcing constraints across color channels to reduce color washing effects. Other methods, such as [17], perform denoising by explicitly modeling the color information at each pixel.

A different approach to treating such correlated color information is through color-space conversion where the information between color spaces can be largely decorrelated. Such an approach was employed in [12] in extending the BM3D algorithm for color images (CBM3D). There, the authors identify similar patches using the luminance channel, to which the human visual system is more sensitive. Denoising is however performed on all channels simultaneously. In general, any grayscale denoising method can be applied to denoising color images through such transformations. However, such color-space conversions corrupt the noise characteristics. Consequently, we perform denoising in the RGB color space, but only the noisy image luminance is used to perform geometric clustering. The parameters for denoising are however individually learned in each color channel.

Fig. 9 illustrates the results obtained by our method with its naive extension to color images. The noisy images are formed by adding simulated 5% WGN in each channel.[9] In terms of the PSNR, the best performing method overall is CBM3D [12]. However, visually, our method is quite comparable with it and is significantly better than that in [21], where there is considerable loss of finer details, and Joshi *et al.* [17], where the denoised images still retain some noise. These results are encouraging

[9]The original and noisy images, along with results for methods in [21] and [17], were obtained from http://research.microsoft.com/en-us/um/redmond/groups/ivm/twocolordeconvolution/supplemental_results/denoising.html. The 5% noise corresponds to a standard deviation of approximately 12 in each color channel.

(a) Barbara (cropped) (b) Noisy (c) K-SVD

(d) NLSM (e) BM3D (f) PLOW

Fig. 7. Denoising of the Barbara image for WGN with $\sigma = 25$. Regions cropped from the images are compared. (f) Our proposed method (MSE of 62.64) produces visually superior results when compared with (c) K-SVD [13] (MSE of 73.95), (d) NLSM [16] (MSE of 61.00), and (e) BM3D [12] (MSE of 56.17). High-resolution images can be viewed at http://users.soe.ucsc.edu/~priyam/PLOW/.

Lena Peppers Boat Stream

Fig. 8. Some benchmark images that we use to perform the quantitative evaluation of method performances.

considering that CBM3D and [17] are specifically designed to handle color images.

Until now, our experiments involved images corrupted by simulated WGN. Although the Gaussian pdf makes a good noise model, real noise is signal dependent [3], [21]. To demonstrate our performance in such situations, we apply our method to denoising some real noisy images with unknown noise characteristics. For these experiments, an estimate of the noise variance was used as an input to our method. The best results optimized using the Q-metric [46] are shown in Fig. 10, where we compare our results to the commercial Neat Image denoising method that specifically handles intensity-dependent noise profiles. Even for such non-Gaussian correlated noise, our method suppresses the noise effectively, while retaining the finer details.

We point out that the parameters used for our method are kept fixed across all noise levels and images. For all our experiments, we use a patch size of $n = 11 \times 11$, with the number of clusters K set to 15. The smoothing parameter, which controls the amount of denoising, is also kept fixed at $h^2 = 1.75\sigma^2 n$. In general, these parameters can be tuned on a per-image basis, manually or using some no-reference image quality measure [46]. In our opinion, such tunable parameters make a method less practical. Results presented in this paper thus use the fixed parameter settings previously mentioned. However, for highly textured images (e.g., boat and stream images), the noise variance tends to be overestimated by (21) when considering strong noise ($\sigma \geqslant 25$). This results in slightly oversmoothened denoised images. For such cases, we provided our algorithm with a lower noise variance. While these parameters influence the performance of the PLOW, other parameters such as those used to compute the LARK, the number of iterations in K-means, and the size of the search window, also exert some effect on the denoising result. However, these are also kept fixed without any need for tuning by the user.

In terms of computational complexity, denoising a 256×256 grayscale image with a nonoptimized (MATLAB and Mex) implementation of our method takes approximately 75 s, with the prefiltering step accounting for about half the processing time. The feature formation and clustering step takes roughly 10%

TABLE I
DENOISING PERFORMANCE OF SOME POPULAR METHODS (NLSM [16] AND BM3D [12]) UNDER WGN CORRUPTION, COMPARED WITH PLOW, WITH AND WITHOUT ORACLE INFORMATION. RESULTS NOTED ARE AVERAGE (TOP) PSNR, (MIDDLE) SSIM [45], AND (BOTTOM) Q- MEASURE [46] OVER FIVE INDEPENDENT NOISE REALIZATIONS FOR EACH σ

σ	**House** (256 × 256)			**Lena** (512 × 512)			**Barbara** (512 × 512)		
	NLSM	**BM3D**	**PLOW**	**NLSM**	**BM3D**	**PLOW**	**NLSM**	**BM3D**	**PLOW**
	39.91	39.80	39.52	**38.72**	**38.73**	**38.66**	**38.46**	38.30	37.98
5	**0.958**	**0.957**	**0.954**	**0.945**	**0.945**	**0.946**	**0.965**	**0.965**	0.946
	42.35	**42.58**	42.20	35.18	**35.37**	34.75	**69.80**	69.60	69.14
	35.27	34.95	34.72	34.17	**34.26**	33.90	32.98	**33.09**	32.17
15	**0.902**	0.890	**0.893**	**0.893**	**0.895**	**0.890**	**0.920**	**0.923**	**0.913**
	36.10	36.37	**36.98**	21.08	21.07	**21.59**	**55.41**	55.25	55.28
	33.14	32.89	32.70	31.84	**32.07**	31.92	30.34	**30.67**	30.20
25	**0.866**	**0.859**	**0.859**	**0.855**	**0.861**	**0.859**	**0.876**	**0.886**	**0.879**
	20.07	20.11	**20.39**	11.43	11.45	**11.69**	**37.87**	37.80	37.72
	28.99	**29.25**	29.08	27.55	**28.58**	28.32	25.68	**26.75**	26.19
50	**0.814**	0.802	0.780	0.774	**0.788**	0.759	0.748	**0.778**	0.755
	–	–	–	–	–	–	–	–	–

σ	**Peppers** (256 × 256)			**Boat** (512 × 512)			**Stream** (512 × 512)		
	NLSM	**BM3D**	**PLOW**	**NLSM**	**BM3D**	**PLOW**	**NLSM**	**BM3D**	**PLOW**
	38.14	**38.06**	37.69	**37.36**	**37.28**	37.24	**35.75**	**35.75**	35.59
5	**0.955**	**0.956**	**0.954**	**0.941**	**0.939**	**0.941**	**0.964**	**0.964**	**0.962**
	76.37	76.17	75.60	37.21	**37.38**	36.95	**31.12**	30.94	30.58
	32.76	32.65	31.82	**32.17**	**32.11**	31.53	**28.88**	28.74	28.71
15	**0.905**	**0.906**	**0.899**	**0.855**	**0.854**	0.840	**0.852**	**0.845**	**0.849**
	64.00	64.02	**64.99**	27.16	27.47	**28.38**	21.51	**22.21**	19.74
	30.06	**30.07**	29.53	**29.73**	**29.83**	29.59	**26.27**	26.14	**26.20**
25	**0.864**	**0.868**	**0.859**	**0.794**	**0.800**	**0.794**	**0.745**	0.735	**0.747**
	49.55	49.87	**50.13**	14.38	**14.49**	14.19	12.18	**12.54**	12.14
	25.16	25.85	**26.32**	25.46	**26.20**	**26.13**	22.43	23.08	**23.38**
50	**0.766**	**0.775**	0.752	0.656	**0.685**	0.674	0.489	0.535	**0.571**
	–	–	–	–	–	–	–	–	–

Noisy images are clipped to lie within the [0–255] intensity range. This, along with the fact that we report the average result of five runs in each case, may result in BM3D and NLSM figures differing from those reported in [12] and [16], respectively. Reliance on detecting anisotropic regions in noisy images makes the Q-measure inapplicable for $\sigma = 50$ cases [46].

Fig. 9. Comparison of the denoising performance for color images corrupted by 5% WGN in each color channel. Methods compared with are CBM3D [12] and the works in [21] (first order) and [17]. Note that CBM3D and the method of Joshi *et al.* are specifically for color images. Images at native resolutions are available at http://users.soe.ucsc.edu/~priyam/PLOW/.

of the execution time, whereas the majority (88%) of the time is spent in estimating the PLOW parameters and filtering the noisy patches. The aggregation step takes negligible time, accounting for the remaining 2% of the time, along with other overheads. Compared with our method, NLSM takes, on average, 170 s to denoise the same images, whereas the optimized

Fig. 10. Denoising of some real noisy color images. Our results are visually comparable with those obtained from the commercial Neat Image denoising method, although Neat Image specifically addresses color images with real noise profiles. Neat Image denoising software is available at http://www.neatimage.com. High-resolution versions of images shown here can be compared at http://users.soe.ucsc.edu/~priyam/PLOW/.

(Mex) code for BM3D is much faster (about 1 s). A simple speedup for our method can be achieved by denoising only every third patch, bringing the average execution time down to approximately 17 s. Although this results in a minor drop of 0.2 dB in the PSNR, the visual differences are almost imperceptible.

V. Summary and Future Work

In this paper, we have proposed a method of denoising motivated from our previous work in analyzing the performance bounds of patch-based denoising methods. We have developed a locally optimal Wiener-filter-based method and have extended it to take advantage of patch redundancy to improve the denoising performance. Our denoising approach does not require parameter tuning and is practical, with the added benefit of a clean statistical motivation and analytical formulation. We analyzed the framework in depth to show its relation to nonlocal means and residual filtering methods such as [50]. Through experimental validation, we have shown that our method produces results quite comparable with the state of the art.

While mainly developed for grayscale images, with trivial modification, our method achieves near state-of-the-art performance in denoising color images as well. The denoising performance can be expected to improve further by taking into account the correlation across color components. Since the method works by learning the moments in geometrically similar patches, the interchannel color dependences can be implicitly captured in this framework. In a more practical setting where signal-dependent noise is observed, the clustering step needs to take into account color (or intensity) information as well. The noise in each cluster can be then assumed to be homogeneous [21], and the proposed filter can be independently applied in each cluster.

Appendix A
Derivation of Expression for $\mathbf{C}_{\underline{\zeta}_i}$

Here, we derive an expression for the covariance matrix $\mathbf{C}_{\underline{\zeta}_i}$ based on the data model [see (10)], i.e.,

$$\underline{\mathbf{y}}_i = \mathbf{A}_i \mathbf{z}_i + \underbrace{\underline{\boldsymbol{\varepsilon}}_i + \underline{\boldsymbol{\eta}}_i}_{\underline{\boldsymbol{\zeta}}_i} \tag{28}$$

where $\underline{\boldsymbol{\varepsilon}}_i = [\ldots \boldsymbol{\varepsilon}_{ij}^T \ldots]^T$ and $\underline{\boldsymbol{\eta}}_i = [\ldots \boldsymbol{\eta}_j^T \ldots]^T$ obtained from all patches $\mathbf{y}_j$ similar to a given $\mathbf{y}_i$. As per definition of $\boldsymbol{\varepsilon}_{ij}$ [see (5)], $\underline{\boldsymbol{\varepsilon}}_i$ and $\underline{\boldsymbol{\eta}}_i$ are independent of each other, which leads to the covariance matrix being

$$\mathbf{C}_{\underline{\zeta}_i} = \mathbf{C}_{\underline{\boldsymbol{\eta}}_i} + \mathbf{C}_{\underline{\boldsymbol{\varepsilon}}_i}. \tag{29}$$

Assuming the noisy patches $\mathbf{y}_i$ and, hence, the noise $\boldsymbol{\eta}_i$ to be i.i.d., the covariance matrix $\mathbf{C}_{\underline{\boldsymbol{\eta}}_i}$ takes the following form:

$$\mathbf{C}_{\underline{\boldsymbol{\eta}}_i} = \sigma^2 \mathbf{I}_q \tag{30}$$

where $\mathbf{I}_q$ is the $q \times q$ identity matrix with dimension $q = nN_i$ dependent on the level of redundancy exhibited by the $\mathbf{y}_i$ patch. Furthermore, assuming that the pixels within a (noise-free) patch $\mathbf{z}_i$ and, as a result, $\boldsymbol{\varepsilon}_{ij}$ are i.i.d., we obtain a diagonal form for $\mathbf{C}_{\underline{\boldsymbol{\varepsilon}}_i}$. The diagonal elements for this matrix can be derived from the definition of (5) as

$$\begin{aligned}
\boldsymbol{\varepsilon}_{ij} &= \mathbf{z}_j - \mathbf{z}_i = (\mathbf{y}_j - \mathbf{y}_i) - (\boldsymbol{\eta}_j - \boldsymbol{\eta}_i) \\
\Rightarrow E\left[\|\boldsymbol{\varepsilon}_{ij}\|^2\right] &= E\left[\|(\mathbf{y}_j - \mathbf{y}_i) - (\boldsymbol{\eta}_j - \boldsymbol{\eta}_i)\|^2\right] \\
&= E\left[\|\mathbf{y}_j - \mathbf{y}_i\|^2\right] + E\left[\|\boldsymbol{\eta}_j - \boldsymbol{\eta}_i\|^2\right] \\
&\quad - 2E\left[(\mathbf{y}_j - \mathbf{y}_i)^T(\boldsymbol{\eta}_j - \boldsymbol{\eta}_i)\right] \\
&= E\left[\|\mathbf{y}_j - \mathbf{y}_i\|^2\right] \\
&\quad - 2\big(E\left[(\mathbf{z}_j - \mathbf{z}_i)^T(\boldsymbol{\eta}_j - \boldsymbol{\eta}_i)\right] \\
&\qquad + E\left[(\boldsymbol{\eta}_j - \boldsymbol{\eta}_i)^T(\boldsymbol{\eta}_j - \boldsymbol{\eta}_i)\right]\big) + 2\sigma^2 n \\
&= E\left[\|\mathbf{y}_j - \mathbf{y}_i\|^2\right] - 2\sigma^2 n
\end{aligned} \tag{31}$$

where the last step assumes the noise to be independent of $\mathbf{z}$. As mentioned before, assuming the $\boldsymbol{\varepsilon}_{ij}$ vectors to be i.i.d., we can write

$$\mathbf{C}_{\boldsymbol{\varepsilon}_{ij}} = \left(\frac{1}{n} E\left[\|\mathbf{y}_j - \mathbf{y}_i\|^2\right] - 2\sigma^2\right)\mathbf{I} \tag{32}$$

$$\Rightarrow \mathbf{C}_{\underline{\boldsymbol{\varepsilon}}_i} = \begin{bmatrix} \ddots & & \mathbf{0} \\ & \mathbf{C}_{\boldsymbol{\varepsilon}_{ij}} & \\ \mathbf{0} & & \ddots \end{bmatrix} \tag{33}$$

from which we obtain the covariance matrix $\mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}$ as

$$\mathbf{C}_{\underline{\boldsymbol{\zeta}}_i} = \mathbf{C}_{\underline{\boldsymbol{\varepsilon}}_i} + \mathbf{C}_{\underline{\boldsymbol{\eta}}_i} = \begin{bmatrix} \ddots & & \mathbf{0} \\ & \delta_{ij}^2 \mathbf{I} & \\ \mathbf{0} & & \ddots \end{bmatrix} \tag{34}$$

where $\delta_{ij}^2 = (1/n)\, E[\|\boldsymbol{\varepsilon}_{ij}\|^2] = (1/n)\, E[\|\mathbf{y}_j - \mathbf{y}_i\|^2] - \sigma^2$. Note that, in the aforementioned expression, the covariance for $\underline{\boldsymbol{\varepsilon}}_i$ and, hence, $\underline{\boldsymbol{\zeta}}_i$ are estimated patchwise, whereas the covariance related to (homogeneous) noise $\underline{\boldsymbol{\eta}}_i$ only varies in dimensionality depending on the redundancy level of the patch under consideration.

Appendix B
Derivation of Redundancy Exploiting Wiener Filter

Here, we derive the LMMSE estimator for the data model in (10). As shown in (11), the LMMSE estimator for each patch can be obtained using its N_i nearest neighbors as

$$\widehat{\mathbf{z}}_i = \bar{\mathbf{z}} + \left(\mathbf{C}_{\mathbf{z}}^{-1} + \mathbf{A}_i^T \mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}^{-1} \mathbf{A}_i\right)^{-1} \mathbf{A}_i^T \mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}^{-1} (\underline{\mathbf{y}}_i - \mathbf{A}_i \bar{\mathbf{z}}) \tag{35}$$

where $\mathbf{A}_i = [\mathbf{I}, \ldots, \mathbf{I}]^T$ is formed by stacking together N_i identity matrices, each of size $n \times n$. With $\mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}$ having a diagonal form [see (12)], we can simplify (35) by noting that

$$\begin{aligned}
&\mathbf{A}_i^T \mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}^{-1} (\underline{\mathbf{y}}_i - \mathbf{A}_i \bar{\mathbf{z}}) \\
&\quad = [\mathbf{I} \ldots \mathbf{I}] \begin{bmatrix} \ddots & & \mathbf{0} \\ & [\delta_{ij}^{-2}\mathbf{I}] & \\ \mathbf{0} & & \ddots \end{bmatrix} \left(\begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{N_i} \end{bmatrix} - \begin{bmatrix} \bar{\mathbf{z}} \\ \vdots \\ \bar{\mathbf{z}} \end{bmatrix} \right) \\
&\quad = \left[\ldots \delta_{ij}^{-2}\mathbf{I} \ldots\right] \begin{bmatrix} \vdots \\ (\mathbf{y}_j - \bar{\mathbf{z}}) \\ \vdots \end{bmatrix} = \sum_{j=1}^{N_i} \delta_{ij}^{-2} (\mathbf{y}_j - \bar{\mathbf{z}})
\end{aligned} \tag{36}$$

$$\begin{aligned}
&\mathbf{A}_i^T \mathbf{C}_{\underline{\boldsymbol{\zeta}}_i}^{-1} \mathbf{A}_i \\
&\quad = \left[\ldots \delta_{ij}^{-2}\mathbf{I} \ldots\right] \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix} = \sum_{j=1}^{N_i} \delta_{ij}^{-2} \mathbf{I}.
\end{aligned} \tag{37}$$

This gives rise to a simplified LMMSE estimator expression

$$\widehat{\mathbf{z}}_i = \bar{\mathbf{z}} + \left(\mathbf{C}_{\mathbf{z}}^{-1} + \sum_{j=1}^{N_i} \delta_{ij}^{-2} \mathbf{I}\right)^{-1} \sum_{j=1}^{N_i} \delta_{ij}^{-2} (\mathbf{y}_j - \bar{\mathbf{z}}) \tag{38}$$

$$= \bar{\mathbf{z}} + \left(\frac{\mathbf{C}_{\mathbf{z}}^{-1}}{\sum_{j=1}^{N_i} \delta_{ij}^{-2}} + \mathbf{I}\right)^{-1} \sum_{j=1}^{N_i} \frac{\delta_{ij}^{-2}}{\sum_{j=1}^{N_i} \delta_{ij}^{-2}} (\mathbf{y}_j - \bar{\mathbf{z}}). \tag{39}$$

Implementing this estimator requires $\mathbf{C}_{\mathbf{z}}$ to be invertible. However, $\mathbf{C}_{\mathbf{z}}$ can be ill conditioned and even rank deficient, leading to an inaccurate estimation of $\mathbf{z}_i$. To alleviate this problem, we make use of the matrix inversion lemma [31] to obtain a form that does not require inversion of $\mathbf{C}_{\mathbf{z}}$:

$$\left(\frac{\mathbf{C}_{\mathbf{z}}^{-1}}{\sum_j \delta_{ij}^{-2}} + \mathbf{I}\right)^{-1} = \mathbf{I} - \left(\sum_j \delta_{ij}^{-2} \mathbf{C}_{\mathbf{z}} + \mathbf{I}\right)^{-1}. \tag{40}$$

This leads to an alternative expression for the LMMSE estimator as

$$\begin{aligned}
\widehat{\mathbf{z}}_i &= \bar{\mathbf{z}} + \left[\mathbf{I} - \left(\sum_j \delta_{ij}^{-2} \mathbf{C}_{\mathbf{z}} + \mathbf{I}\right)^{-1}\right] \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} (\mathbf{y}_j - \bar{\mathbf{z}}) \\
&= \bar{\mathbf{z}} + \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} (\mathbf{y}_j - \bar{\mathbf{z}}) \\
&\quad - \left(\sum_j \delta_{ij}^{-2} \mathbf{C}_{\mathbf{z}} + \mathbf{I}\right)^{-1} \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} (\mathbf{y}_j - \bar{\mathbf{z}}) \\
&= \sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} \left[\mathbf{y}_j - \left(\sum_j \delta_{ij}^{-2} \mathbf{C}_{\mathbf{z}} + \mathbf{I}\right)^{-1} (\mathbf{y}_j - \bar{\mathbf{z}})\right] \\
&= \left[\sum_j \frac{\delta_{ij}^{-2} \mathbf{y}_j}{\sum_j \delta_{ij}^{-2}}\right] \\
&\quad - \left[\sum_j \frac{\delta_{ij}^{-2}}{\sum_j \delta_{ij}^{-2}} \left(\sum_j \delta_{ij}^{-2} \mathbf{C}_{\mathbf{z}} + \mathbf{I}\right)^{-1} (\mathbf{y}_j - \bar{\mathbf{z}})\right].
\end{aligned} \tag{41}$$

Note that the first part of the aforementioned formulation is closely related to NLM [8], [9] (using weights $w_{ij} = \delta_{ij}^{-2}$) with an added term that processes the residuals between the noisy patches and the estimated mean patch.

APPENDIX C
DERIVATION OF APPROXIMATE SIMILARITY MEASURE

In Section II-B, we derived an extension for the Wiener filter where photometrically similar patches contribute in denoising a given reference patch. This was analyzed through a modified per-patch data model. In the final filter formulation, the contributing weights of photometrically similar patches were determined to be $w_{ij} = \delta_{ij}^{-2}$, where

$$\delta_{ij}^2 = \frac{1}{n} E\left[\|\mathbf{z}_i - \mathbf{z}_j\|^2\right] + \sigma^2. \tag{42}$$

However, computing δ_{ij} for a pair of $\mathbf{z}_i$ and $\mathbf{z}_j$ random vectors is not possible given only the two observations. Furthermore, this would require access to the noise-free image. In practice, δ_{ij} and, hence, w_{ij} need to be estimated from the corresponding noisy observations $\mathbf{y}_i$ and $\mathbf{y}_j$, respectively. However, let us first assume that the noise-free patches are made available to us. Next, we show that the weight formulation employed in (23) is simply an approximation that can be derived from (42).

Let us rewrite (23) in terms of the original noise-free image patches as

$$w_{ij} = \frac{1}{\sigma^2} \exp\left\{-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{h^2}\right\} \tag{43}$$

$$\Rightarrow \delta_{ij}^2 = w_{ij}^{-1} = \sigma^2 \exp\left\{\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{h^2}\right\}. \tag{44}$$

Note that the aforementioned equation would be the ideal weights that we *estimate* using the noisy observations $\mathbf{y}_i$ and $\mathbf{y}_j$ in (23). Let us define $u = \|\mathbf{z}_i - \mathbf{z}_j\|^2/h^2$. Since we consider only photometrically similar patches that satisfy the condition in (5), we know that $\|\mathbf{z}_i - \mathbf{z}_j\|^2 \leqslant \gamma^2 \ll \sigma^2 n$ [1]. Thus, by choosing $h^2 \geqslant \sigma^2 n$, we can guarantee that $u < 1$. Consequently, as h^2 increases, u approaches 0. We can then write the Taylor expansion of the exponential function around $u = 0$ as

$$\begin{aligned} e^u &= 1 + u + O(u^2) \approx 1 + u \quad [\text{since } u < 1] \\ &= 1 + \frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{\sigma^2 n} \\ \Rightarrow \delta_{ij}^2 &= \sigma^2 e^u \approx \sigma^2 + \frac{1}{n}\|\mathbf{z}_i - \mathbf{z}_j\|^2. \end{aligned} \tag{45}$$

Comparing (42) with the above expression, it is easy to observe their similarities. As mentioned earlier, the expected value of (42) cannot be accurately calculated from a single pair of $\mathbf{z}_i$ and $\mathbf{z}_j$ observations. As a result, it is dropped when computing the weights.

The aforementioned derivation assumed the knowledge of the distance between the noise-free similar patches. As such, these are the "oracle" weights that we would ideally want to use for denoising. However, in practice, only the noisy patches are observed. As a result, the actual weight function is approximated by replacing the noise-free $\mathbf{z}_i$ and $\mathbf{z}_j$ patches with their corresponding noisy $\mathbf{y}_i$ and $\mathbf{y}_j$ observations, giving rise to the expression in (23) as

$$w_{ij} = \frac{1}{\sigma^2} \exp\left\{-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{h^2}\right\}. \tag{46}$$

Note that the distance between noisy patches can be much higher than those between the underlying noise-free patches. As a result, a larger smoothing term is needed for denoising. In our paper, we set $h^2 = 1.75\sigma^2 n$ for all noise levels and images.

REFERENCES

[1] P. Chatterjee and P. Milanfar, "Is denoising dead?," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 895–911, Apr. 2010.

[2] P. Chatterjee and P. Milanfar, "Practical bounds on image denoising: From estimation to information," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1221–1233, May 2011.

[3] G. E. Healey and R. Kondepudy, "Radiometric CCD camera calibration and noise estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 3, pp. 267–276, Mar. 1994.

[4] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. Int. Conf. Comput. Vis.*, Washington, DC, Jan. 1998, pp. 839–846.

[5] S. M. Smith and J. M. Brady, "SUSAN—A new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, May 1997.

[6] L. P. Yaroslavsky, *Digital Picture Processing*. Secaucus, NJ: Springer-Verlag, 1985.

[7] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

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

[9] 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.

[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] C. Kervrann and J. Boulanger, "Local adaptivity to variable smoothness for exemplar-based image denoising and representation," *Int. J. Comput. Vis.*, vol. 79, no. 1, pp. 45–69, Aug. 2008.

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

[13] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[14] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[15] P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438–1451, Jul. 2009.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 2272–2279.

[17] N. Joshi, C. L. Zitnick, R. Szeliski, and D. Kriegman, "Image deblurring and denoising using color priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Miami, FL, Jun. 2009, pp. 1550–1557.

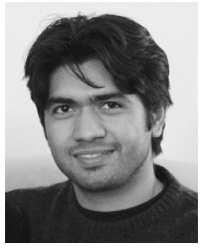
[18] D. D. Muresan and T. W. Parks, "Adaptive principal components and image denoising," in *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, Sep. 2003, vol. 1, pp. 101–104.

[19] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, no. 4, pp. 1531–1549, Apr. 2010.

[20] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, Apr. 2009.

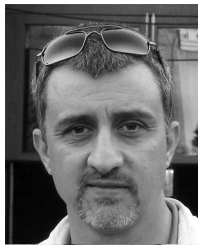
[21] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 299–314, Feb. 2008.

[22] S. G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1522–1531, Sep. 2000.
[23] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using a scale mixture of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.
[24] S. Lyu and E. P. Simoncelli, "Modeling multiscale subbands of photographic images with fields of Gaussian scale mixtures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 693–706, Apr. 2009.
[25] F. Luisier, T. Blu, and M. Unser, "A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 593–606, Mar. 2007.
[26] H. Cramér, *Mathematical Methods of Statistics*. Princeton, NJ: Princeton Univ. Press, 1946.
[27] C. R. Rao, "Information and the accuracy attainable in the estimation of statistical parameters," *Bull. Calcutta Math. Soc.*, vol. 37, pp. 81–89, 1945.
[28] C. R. Rao, "Minimum variance and the estimation of several parameters," in *Proc. Camb. Philos. Soc.*, 1946, vol. 43, pp. 280–283.
[29] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, ser. Signal Processing. Upper Saddle River, NJ: Prentice-Hall, 1993.
[30] M. Shao, K. E. Barner, and R. C. Hardie, "Partition-based interpolation for color filter array demosaicking and super-resolution reconstruction," *Opt. Eng.*, vol. 44, no. 10, p. 107 003, Oct. 2005.
[31] M. A. Woodbury, Inverting modified matrices Stat. Res. Group, Princeton Univ., Princeton, NJ, Memorandum Rep. 42, 1950.
[32] G. Yu, G. Sapiro, and S. Mallat, "Image modeling and enhancement via structured sparse model selection," in *Proc. IEEE Int. Conf. Image Process.*, , Hong Kong, Sep. 2010, pp. 1641–1644.
[33] H. J. Seo and P. Milanfar, "Training-free, generic object detection using locally adaptive regression kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1688–1704, Sep. 2010.
[34] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[35] B. Efron, "Bootstrap methods: Another look at the jacknife," *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, Jan. 1979.
[36] Y. Chen, A. Wiesel, Y. C. Eldar, and A. O. Hero, III, "Shrinkage algorithms for MMSE covariance estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5016–5029, Oct. 2010.
[37] S. Kritchman and B. Nadler, "Non-parametric detection of the number of signals, hypothesis testing and random matrix theory," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3930–3941, Oct. 2009.
[38] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.
[39] V. Duval, J.-F. Aujol, and Y. Gousseau, On the parameter choice for the non-local means Mar. 2010 [Online]. Available: http://hal.archives-ouvertes.fr/docs/00/46/88/56/PDF/nlmeans2.pdf, CMLA Preprint
[40] A. C. Aitken, "On least squares and linear combinations of observations," *Proc. R. Soc. Edinb.*, vol. 55, pp. 42–48, 1935.
[41] J. Salmon and Y. Strozecki, "From patches to pixels in non-local methods: Weighted-average reprojection," in *Proc. IEEE Int. Conf. Image Process.*, , Hong Kong, Sep. 2010, pp. 1929–1932.
[42] C. Angelino, E. Debreuve, and M. Barlaud, "Patch confidence k-nearest neighbors denoising," in *Proc. IEEE Conf. Image Process.*, , Hong Kong, Sep. 2010, pp. 1129–1132.
[43] S. Kritchman and B. Nadler, "Determining the number of components in a factor model from limited noisy data," *Chemometrics Intell. Lab. Syst.*, vol. 94, no. 1, pp. 19–32, Nov. 2008.
[44] P. Chatterjee and P. Milanfar, "Learning denoising bounds for noisy images," in *Proc. IEEE Int. Conf. Image Process.*, , Hong Kong, Sep. 2010, pp. 1157–1160.
[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[46] X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3116–3132, Dec. 2010.
[47] D. Zoran and Y. Weiss, "Scale invariance and noise in natural images," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 2209–2216.
[48] R. W. G. Hunt, *The Reproduction of Colour*. Hoboken, NJ: Wiley, Nov. 2005.
[49] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.
[50] D. Brunet, E. R. Vrscay, and Z. Wang, "The use of residuals in image denoising," in *Proc. Int. Conf. Image Anal. Recog.*, Halifax, NS, Canada, Jul. 2009, pp. 1–12.



Priyam Chatterjee (S'07) received the B.Tech. degree in information technology from the University of Kalyani, Kalyani, India, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, and the Ph.D. degree in electrical engineering from the University of California, Santa Cruz, in 2011.

He was an Intern with Microsoft Research, Redmond, WA, in 2010 and is currently with Pelican Imaging Corporation, Mountain View, CA. His research interests are in image and video processing (denoising, interpolation, deblurring, and super-resolution).



Peyman Milanfar (F'10) received the B.S. degree in electrical engineering and mathematics from the University of California, Berkeley, and the M.S. and Ph.D. degrees in electrical engineering from Massachusetts Institute of Technology, Cambridge.

Until 1999, he was a Senior Research Engineer with SRI International, Menlo Park, CA. He is a Professor of electrical engineering and the Associate Dean for research with Baskin School of Engineering, University of California, Santa Cruz. From 1998 to 2000, he was a Consulting Assistant Professor of computer science with Stanford University, Stanford, CA, where he was also a Visiting Associate Professor in 2002. His research interests include statistical signal, image and video processing, and computational photography and vision.

Dr. Milanfar was a recipient of a National Science Foundation CAREER Award in 2000 and the Best Paper Award from the IEEE Signal Processing Society in 2010. He is a member of the IEEE Signal Processing Society's Awards Board and the Image, Video, and Multidimensional Signal Processing technical committee.