Global Image Denoising

Hossein Talebi, *Student Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

***Abstract*— Most existing state-of-the-art image denoising algorithms are based on exploiting similarity between a relatively modest number of patches. These patch-based methods are strictly dependent on patch matching, and their performance is hamstrung by the ability to reliably find sufficiently similar patches. As the number of patches grows, a point of diminishing returns is reached where the performance improvement due to more patches is offset by the lower likelihood of finding sufficiently close matches. The net effect is that while patch-based methods, such as BM3D, are excellent overall, they are ultimately limited in how well they can do on (larger) images with increasing complexity. In this paper, we address these shortcomings by developing a paradigm for truly global filtering where each pixel is estimated from all pixels in the image. Our objectives in this paper are two-fold. First, we give a statistical analysis of our proposed global filter, based on a spectral decomposition of its corresponding operator, and we study the effect of truncation of this spectral decomposition. Second, we derive an approximation to the spectral (principal) components using the Nyström extension. Using these, we demonstrate that this global filter can be implemented efficiently by sampling a fairly small percentage of the pixels in the image. Experiments illustrate that our strategy can effectively globalize any existing denoising filters to estimate each pixel using all pixels in the image, hence improving upon the best patch-based methods.**

***Index Terms*— Image denoising, non-local filters, Nyström extension, spatial domain filter, risk estimator.**

I. Introduction

DENOISING of images is perhaps the most basic image restoration problem. The degradation model for the denoising problem can be described as:

$$\mathbf{y} = \mathbf{z} + \mathbf{e} \tag{1}$$

where column vectors $\mathbf{z}$ and $\mathbf{y}$ denote the (vectorized) underlying latent image and its noisy observation, respectively. The vector $\mathbf{e}$ represents zero-mean white noise[1] with variance σ^2 (which is assumed to be spatially invariant in this paper). There have been numerous denoising algorithms to estimate $\mathbf{z}$ from $\mathbf{y}$, and in general most of these methods can be categorized as patch-based filters. While some denoising approaches such as the bilateral filter [1], LARK [2] and NLM [3] estimate each pixel separately fusing other "similar" neighborhood pixels; more recent state-of-the-art patch-based methods such as BM3D [4] and PLOW [5] denoise a group of similar patches together. Overall, most existing approaches, including the present one, can be unified as data-dependent filtering schemes [6].

Patch-based filtering is founded on the assumption that the latent image has a locally sparse representation in some transform domain. Wavelet and DCT in [4], principle component analysis (PCA) in [7], and over-complete dictionaries in [8] are the frequently used transforms. The filtering process is defined as applying a shrinkage function to the transform coefficients and recovering the estimated patches by inverse transform. However, performance of these patch-based methods is strictly dependent on how well the similar patches are matched [9]. Specifically, for images that are well represented by locally sparse transform (i.e. images with locally repetitive structure such as House in Fig. 6), the shrinkage operator keeps most of the basis elements belonging to the latent signal and effectively removes the noise components. Yet, when the similar patches are not easily representable in a sparse way (i.e. images with locally non-repetitive, or semistochastic structures such as Mandrill in Fig. 6), the signal components and the noise elements can be mistakenly shrunk together. Consequently, performance of the patch-based filtering will be affected by the lack of locally (in the nearest neighbor sense) similar patches [10].

As shown in [6], a spatial domain denoising process has a transform domain filtering interpretation, where the orthogonal basis elements and the shrinkage coefficients are respectively the eigenvectors and eigenvalues of a symmetric, positive definite (data-dependent) filter matrix. For filters such as NLM and LARK the eigenvectors corresponding to the dominant eigenvalues could well represent latent image contents. Based on this idea, the SAIF method [11] was recently proposed which is capable of controlling the denoising strength locally for any given spatial domain method. SAIF iteratively filters local image patches, and the iteration method and iteration number are automatically optimized with respect to locally estimated MSE. Although this algorithm does not set any theoretical limitation over this local window size, computational burden of building a matrix filter for a window as large as the whole image is prohibitively high.

As shown by Williams and Seeger [12], the Nyström method [13] gives a practical solution when working with huge affinity (similarity) matrices by operating on only a small portion of the complete matrix to produce a low-rank approximation. The Nyström method was initially introduced as a technique for finding numerical solutions to eigen-decomposition problems in [13] and [14]. The Nyström extension has been useful for different applications such as manifold learning [15], image segmentation [16], and image

Manuscript received May 2, 2013; revised October 11, 2013; accepted November 11, 2013. Date of publication December 3, 2013; date of current version December 24, 2013. This work was supported in part by the AFOSR 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. Aleksandra Pizurica.

The authors are with the Department of Electrical Engineering, University of California, Santa Cruz, CA 95064 USA (e-mail: htalebi@soe.ucsc.edu; 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.2013.2293425

[1]We make no other distributional assumptions on the noise.

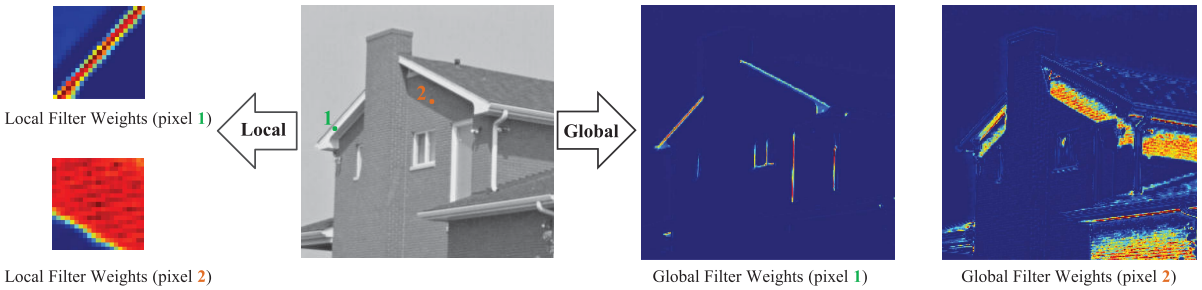


Fig. 1. Comparison of the local and global filter weights for the NLM kernel [3]. The filter weights are computed for the two labeled pixels.

editing [17]. Fortunately, in our global filtering framework, the filter matrix is not a full-rank local filter and thus can be closely approximated with a low-rank matrix using the Nyström method.

Our contribution to this line of research is to introduce an innovative global denoising filter, which takes into account all informative parts of an image (see Fig. 1). Distinctly, with this global filter in hand, the concept of patch-based processing is no longer restrictive, and we are able to show that performance of the existing patch-based filters are improvable.

The block diagram of the proposed global image denoising (GLIDE) framework is illustrated in Fig. 2. As can be seen, after applying a pre-filter on the noisy image, a small fraction of the pixels are sampled to be fed to the Nyström method. Then, the global filter is approximated through its eigenvalues and eigenvectors. The final estimate of the image is constructed by means of shrinkage of the filter eigenvalues.

The paper is organized as follows. Section II describes our statistical analysis of the global filter based on an iterative shrinkage strategy. Section III provides our proposed three-step algorithm for approximating the symmetric global filter based on the Nyström extension. After this eigen-decomposition approximation, performance of our global filter is presented in Section IV. Finally we conclude this paper in Section V.

II. Statistical Analysis of the Filter

With z_i representing the i-th underlying pixel, our measurement model for the denoising problem is:

$$y_i = z_i + e_i, \qquad \text{for} \quad i = 1, \ldots, n, \tag{2}$$

where y_i is the noisy pixel value and e_i denotes the additive noise. The vectorized measurement model for recovering the underlying pixels $\mathbf{z} = [z_1, \ldots, z_n]^T$ is given by (1).

Most spatial domain filters can be represented through the following non-parametric restoration framework [2], [6]:

$$\widehat{z}_i = \arg\min_{z_i} \sum_{j=1}^{n} [z_i - y_j]^2 K_{ij}, \tag{3}$$

where the kernel function K_{ij} measures the similarity between the samples y_i and y_j, and $\widehat{z}_i$ denotes the i-th estimated pixel.

The bilateral filter [1] is perhaps the most well-known kernel which smooths images by means of a nonlinear combination of nearby image values. This filter combines pixel values based on both their geometric closeness and their photometric similarity. The NLM [3] is another very popular data-dependent filter which closely resembles the bilateral filter except that the photometric similarity is captured in a patch-wise manner. More recently, the LARK [2] was introduced which exploits the geodesic distance based on estimated gradients.

Minimizing equation (3) gives a normalized weighted averaging process in which some data-adaptive weights are assigned to each pixel:

$$\widehat{z}_i = \mathbf{w}_i^T \mathbf{y}, \tag{4}$$

where the weight vector $\mathbf{w}_i$ is

$$\mathbf{w}_i = \frac{1}{\sum_{j=1}^{n} K_{ij}} [K_{i1}, K_{i2}, \ldots, K_{in}]^T. \tag{5}$$

in which $[K_{i1}, K_{i2}, \ldots, K_{in}]$ denotes the i-th row of the symmetric kernel matrix $\mathbf{K}$. The filtering process for all the pixels can be represented by stacking the weight vectors together:

$$\widehat{\mathbf{z}} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_n^T \end{bmatrix} \mathbf{y} = \mathbf{W}\mathbf{y}, \tag{6}$$

where the positive, row-stochastic filter matrix $\mathbf{W}$ is used to estimate the denoised signal $\widehat{\mathbf{z}}$. $\mathbf{W}$ is not generally symmetric, though it has real, positive eigenvalues [6]. In particular, the eigenvalues of $\mathbf{W}$ satisfy $0 \leq \lambda_i \leq 1$; the largest one is uniquely equal to one ($\lambda_1 = 1$) while the corresponding eigenvector is $\mathbf{v}_1 = \frac{1}{\sqrt{n}}[1, 1, \ldots, 1]^T$ [18], [19]. The last property implies the desirable feature that a flat image stays unchanged after filtering by $\mathbf{W}$.

While $\mathbf{W}$ is not a symmetric matrix, it can be closely approximated with a symmetric, positive definite, doubly (i.e., row- *and* column-) stochastic matrix [20]. The symmetric $\mathbf{W}$ enables us to compute its eigen-decomposition as follows:

$$\mathbf{W} = \mathbf{V}\mathbf{S}\mathbf{V}^T, \tag{7}$$

in which the eigenvectors $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ specify a complete orthonormal basis for $\mathbb{R}^n$ and $\mathbf{S} = \text{diag}[\lambda_1, \ldots, \lambda_n]$ contains the eigenvalues in decreasing order $0 \leq \lambda_n \leq \cdots < \lambda_1 = 1$.

Denoting n as the total number of the pixels in the image, our one-shot, global filter for the whole image can be expressed as:

$$\widehat{\mathbf{z}} = \mathbf{W}\mathbf{y} = \mathbf{V}\mathbf{S}\mathbf{V}^T\mathbf{y}, \tag{8}$$

This implies that the image $\mathbf{y}$ is first projected onto the eigenvectors of $\mathbf{W}$, then each mode of the projected signal

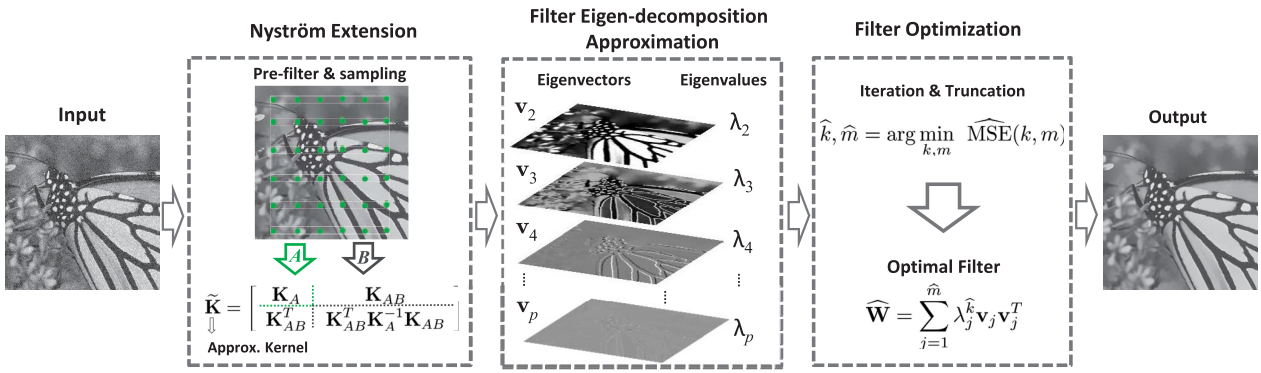


Fig. 2. GLIDE's pipeline. From left to right, for a noisy image we first apply a pre-filter to reduce the noise level. Then using a spatially uniform sampling, the global kernel is approximated by employing the Nyström extension (A and B represent the samples and the rest of the pixels in the image, respectively). As is discussed in Section III, using the obtained kernel, the leading eigenvalues and eigenvectors of the filter are approximated (The eigenvector $\mathbf{v}_1$ is not shown because it is constant). Finally, the optimal filter is constructed by shrinking (iteration and truncation) the eigenvalues. The filter optimization step is detailed in Section II.

is shrunk by its corresponding eigenvalue, and finally after mapping back to the signal domain, the recovered signal $\widehat{\mathbf{z}}$ is produced.

Not surprisingly, the computational burden of constructing and decomposing such a large matrix as $\mathbf{W}$ is prohibitively high. However, the Nyström approximation, combined with our statistical analysis allows an efficient solution. Before proceeding to the filter approximation, the behavior of the filter (in terms of MSE) is analyzed.

A. Ideal Full Space Filter

Let's assume that all of the eigenmodes of the filter $\mathbf{W}$ are used without any change and the filter $\mathbf{W}$ is stochastically independent from the input image $\mathbf{y}$. Then, starting from MSE of each pixel, in Appendix A we show that the overall ideal MSE for the whole image is:

$$\mathrm{MSE} \approx \sum_{j=1}^{n} (1-\lambda_j)^2 b_j^2 + \sigma^2 \lambda_j^2 \tag{9}$$

where in the above, $\|\mathrm{bias}(\widehat{\mathbf{z}})\|^2 = \sum_{j=1}^{n}(1-\lambda_j)^2 b_j^2$ and $\mathrm{var}(\widehat{\mathbf{z}}) = \sigma^2 \sum_{j=1}^{n} \lambda_j^2$ and $\mathbf{b} = \mathbf{V}^T\mathbf{z} = [b_1, \ldots, b_n]^T$ contains the projected signal in all modes (consistent with the MSE analysis in [6]). Apparently MSE is a function of the latent signal, noise, filter eigenvalues and eigenvectors. The filter eigenvalues are the shrinkage factors which directly tune the filtering performance.

As discussed in [6], minimization of the MSE as a function of the filter eigenvalues leads to the Wiener filter:

$$\lambda_j^* = \frac{1}{1+\mathrm{snr}_j^{-1}} \tag{10}$$

where $\mathrm{snr}_j = \frac{b_j^2}{\sigma^2}$. This optimum shrinkage requires exact knowledge of the signal-to-noise ratio in each channel.

Estimation accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we may sacrifice some bias to reduce the variance of the estimated values, and hence may improve the overall estimation accuracy.

B. Truncated Filter

The filtering framework in (8) can be performed for the leading (say $m < n$) eigenvalues of the filter $\mathbf{W}$. As we show in Appendix B, such a filter has the following MSE:

$$\mathrm{MSE}(m) = \sum_{i=1}^{n} z_i^2 + \sum_{j=1}^{m} \Big((\lambda_j^2 - 2\lambda_j) b_j^2 + \sigma^2 \lambda_j^2 \Big) \tag{11}$$

where in the above, $\|\mathrm{bias}(\widehat{\mathbf{z}})\|^2 = \sum_{i=1}^{n} z_i^2 + \sum_{j=1}^{m}(\lambda_j^2 - 2\lambda_j) b_j^2$ and $\mathrm{var}(\widehat{\mathbf{z}}) = \sigma^2 \sum_{j=1}^{m} \lambda_j^2$. For the sake of comparison, we can assume that all the signal modes are available and then $\sum_{i=1}^{n} z_i^2$ can be replaced with $\sum_{i=j}^{n} b_j^2$. After some simplifications we can rewrite our MSE expression as:

$$\mathrm{MSE}(m) = \underbrace{\sum_{j=1}^{n} (1-\lambda_j)^2 b_j^2 + \sigma^2 \lambda_j^2}_{\mathrm{MSE}} + \sum_{j=m+1}^{n} (2\lambda_j - \lambda_j^2) b_j^2 - \sigma^2 \lambda_j^2 \tag{12}$$

As can be seen, $\mathrm{MSE}(m)$ of the truncated filter differs from (9) by the amount given in the second term of (12); i.e. $\Delta\mathrm{MSE} = \sum_{j=m+1}^{n} (2\lambda_j - \lambda_j^2) b_j^2 - \sigma^2 \lambda_j^2$. This difference is also composed of bias and variance parts as $\Delta\mathrm{MSE} = \Delta\|\mathrm{bias}\|^2 + \Delta\mathrm{var}$ where

$$\Delta\|\mathrm{bias}\|^2 = \sum_{j=m+1}^{n} (2\lambda_j - \lambda_j^2) b_j^2 \tag{13}$$

$$\Delta\mathrm{var} = -\sum_{j=m+1}^{n} \sigma^2 \lambda_j^2 \tag{14}$$

This shows, consistent with intuition, that the truncation lowers the variance and increases the bias. Given this analysis, we can determine when truncation improves the MSE. That is, when is $\Delta\mathrm{MSE} < 0$. A simple sufficient condition is that for all j,

$$\mathrm{snr}_j < \frac{\lambda_j}{2-\lambda_j} \tag{15}$$

Intuitively we can conclude that all the channels in the range of $m+1 \le j \le n$ with sufficiently small signal-to-noise ratio should be set to zero. This inequality can also be expressed as:

$$\lambda_j > \frac{2}{1+\mathrm{snr}_j^{-1}} \tag{16}$$

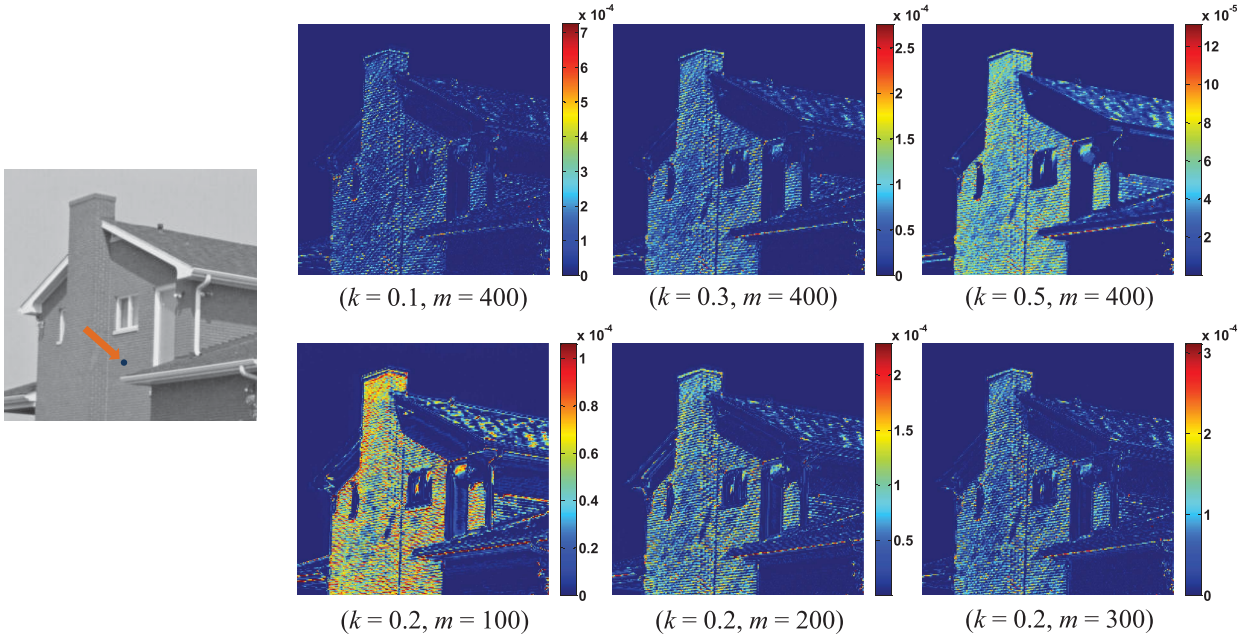


Fig. 3. Filter weights with different shrinkage (k) and truncation (m) parameters are computed for the labeled pixel in the House image.

Comparing this inequality with the Wiener shrinkage criterion in (10), it can be seen that $\lambda_j > 2\lambda_j^*$. That is, the condition implied by (16) is a stronger form of shrinkage than what the Wiener condition would dictate.

C. Iterative Filter

Although the estimated MSE can be reduced by truncating some of the eigenmodes, hard thresholding prevents the accuracy of the estimation to be close to optimal. To ameliorate this shortcoming, iteration can gradually tune the (truncated) filter to softly vary its filtering strength. As such, the iteration and truncation numbers are the only parameters to be globally optimized. Our iterative diffusion model [6] is:

$$\widehat{\mathbf{z}} = \widehat{\mathbf{W}}\mathbf{y} = \mathbf{V}_m \mathbf{S}_m^k \mathbf{V}_m^T \mathbf{y}, \tag{17}$$

where $\mathbf{V}_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$, $\mathbf{S}_m^k = \text{diag}[\lambda_1^k, \lambda_2^k, \ldots, \lambda_m^k]$ and k denotes the iteration number.[2] Fig. 3 illustrates corresponding filter weights of the marked pixel in the House image. As can be seen, different iteration and truncation numbers can effectively vary the behavior of the filter to find similar pixels all over the image.

With this model, we can rewrite (11) for the truncated iterative filter:

$$\text{MSE}(k, m) = \sum_{i=1}^{n} z_i^2 + \sum_{j=1}^{m} \left((\lambda_j^{2k} - 2\lambda_j^k) b_j^2 + \sigma^2 \lambda_j^{2k} \right) \tag{18}$$

Overall, our minimization problem will be extended to estimating the shrinkage ($\widehat{k}$) and truncation ($\widehat{m}$) factors from an estimate of MSE:

$$\widehat{k}, \widehat{m} = \arg\min_{k,m} \widehat{\text{MSE}}(k, m) \tag{19}$$

where $\widehat{\text{MSE}}(k, m)$ denotes an estimate of $\text{MSE}(k, m)$. The shrinkage and truncation parameters are simultaneously optimized such that $(\widehat{k}, \widehat{m})$ is the global minimum of $\widehat{\text{MSE}}(k, m)$. Minimization of $\widehat{\text{MSE}}(k, m)$ determines the best parameters to help avoid under- or over-smoothing.

Although the diffusion iteration is chosen in our framework, our analysis makes it possible to use other iterations too. In general, any iterative approach can be defined as substituting the eigenvalues λ_j with a shrinkage function $f_k(\lambda_j)$ where in the case of diffusion $f_k(\lambda_j) = \lambda_j^k$. Another alternative can be the *boosting* iteration which is a complementary mechanism to recycle lost details of the filtered signal [6], [11]. In this case, the eigenvalues will be shrunk as $f_k(\lambda_j) = 1 - (1 - \lambda_j)^{k+1}$.

D. Practical Filtering

In the estimation of the MSE in (18), we assumed that the filter $\mathbf{W}$ is stochastically independent from the input image $\mathbf{y}$. It has been shown that in the case of a smooth filter (kernel with small gradient), a pre-filter can effectively decouple $\mathbf{W}$ from $\mathbf{y}$ [20]. The smoothness of the filter is approximately true when the filter is computed for locally homogenous patches. By approximating the local signal-to-noise ratio, this type of MSE estimator has been shown to work quite well [11]. However, in the case of the global filter, pixels with different local structures are connected to each other; which means drastic changes in the filter values. In other words, $\mathbf{W}$ and $\mathbf{y}$ are not stochastically decoupled. This nonlinearity specifically affects the estimated variance in the ideal MSE presented in (18). Inspired by the SURE estimator [21], we can show

[2]It is noteworthy that the spectral decomposition of $\mathbf{W}^k$ makes it possible to replace k with any real number. It is important to note that because of this, k can really be thought of as a shrinkage parameter that controls the rate of decay of the modified eigenvalues λ_j^k. Going forward, we will use the terms "shrinkage factor" and "iteration number" interchangeably.

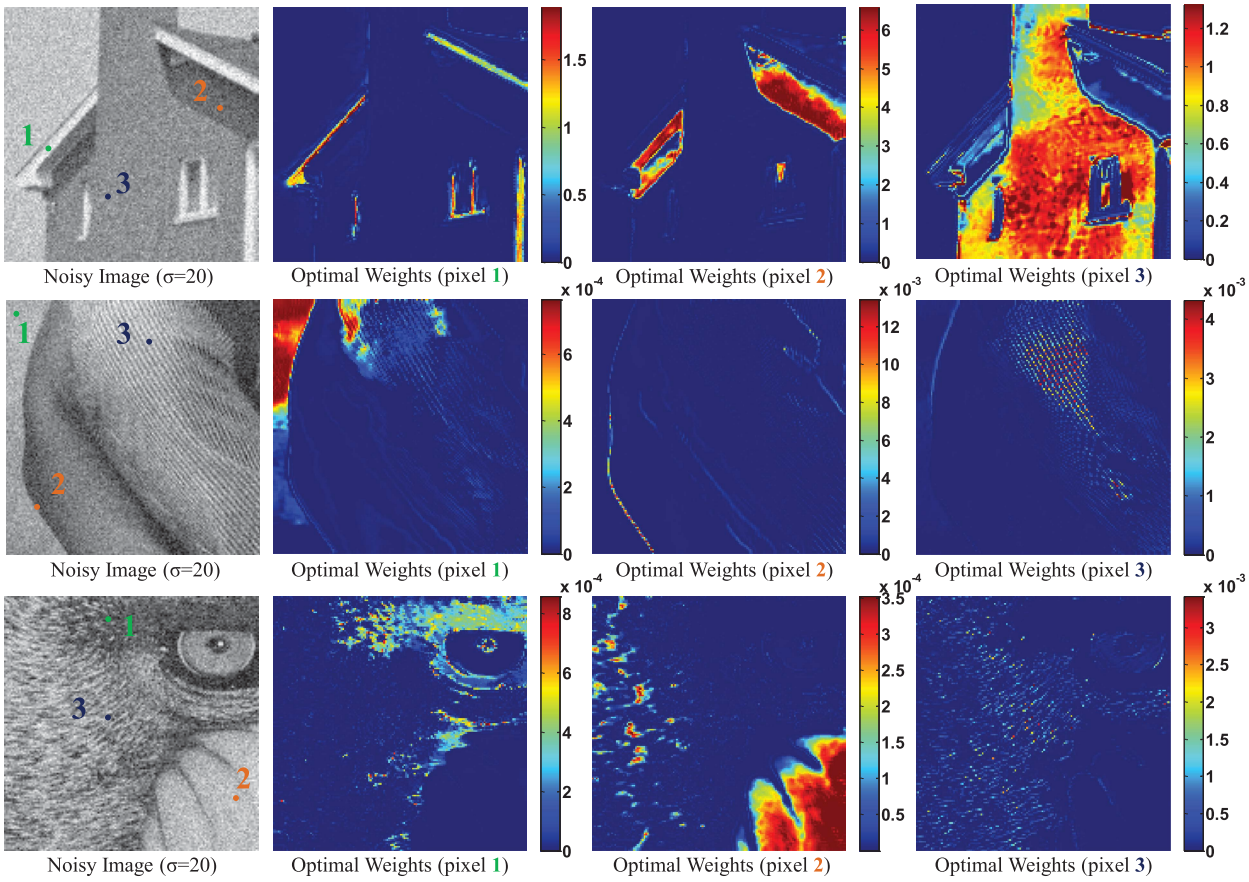


Fig. 4. Optimal filter weights for the labeled pixels in the images. The optimal iteration and truncation numbers for each image are estimated as, House: $\widehat{k} = 0.16$ and $\widehat{m} = 40$, Barbara: $\widehat{k} = 0.14$ and $\widehat{m} = 65$, Mandrill: $\widehat{k} = 0.33$, $\widehat{m} = 165$.

that the estimated variance can be modified as:

$$\text{var}(\widehat{\mathbf{z}}) \approx \sigma^2 \sum_{j=1}^{m} \lambda_j^{2k} + 2\sigma^2 \left(\text{div}(\widehat{\mathbf{z}}(\mathbf{y})) - \sum_{j=1}^{m} \lambda_j^k \right) \quad (20)$$

where $\text{div}(\widehat{\mathbf{z}}(\mathbf{y})) \equiv \sum_i \frac{\partial \widehat{z}_i(\mathbf{y})}{\partial y_i}$. In the case of a strictly linear filter, $\text{div}(\widehat{\mathbf{z}}(\mathbf{y})) = \sum_{j=1}^{m} \lambda_j^k$, which leads to the ideal variance in (18). Intuitively, the second term in (20) takes care of the variance due to the nonlinearity.

It is quite straightforward to show that the bias term in (18) can be better estimated as:

$$\|\text{bias}(\widehat{\mathbf{z}})\|^2 \approx \sum_{i=1}^{n} y_i^2 - \sigma^2 + \sum_{j=1}^{m} (\lambda_j^{2k} - 2\lambda_j^k)(\check{b}_j^2 - \sigma^2) \quad (21)$$

where $\check{\mathbf{b}} = \mathbf{V}^T \mathbf{y}$. Expected value of this estimator is exactly the ideal squared bias in (18). Overall, the estimated MSE of the general nonlinear filter has the following form:

$$\widehat{\text{MSE}}(k,m) = \text{SURE}(k,m) = \sum_{i=1}^{n} y_i^2 - \sigma^2 + \sum_{j=1}^{m} (\lambda_j^{2k} - 2\lambda_j^k)\check{b}_j^2 + 2\sigma^2 \text{div}(\widehat{\mathbf{z}}(\mathbf{y})) \quad (22)$$

which is the SURE estimator [21]. For large data sets (such as our global framework), closeness of the SURE risk estimator to the actual MSE is assured by the law of large numbers.

Approximation of $\text{div}(\widehat{\mathbf{z}}(\mathbf{y}))$ has been studied in [22] and [23]. Ramani's Monte-Carlo algorithm [23] uses a first-order difference approximation to obtain an estimate of the divergence term. Based on this method, the divergence term can be computed from $\text{div}(\widehat{\mathbf{z}}(\mathbf{y})) = \frac{1}{\epsilon}\mathbf{a}^T(\widehat{\mathbf{z}}(\mathbf{y}) - \widehat{\mathbf{z}}(\mathbf{y}'))$ where $\mathbf{y}' = \mathbf{y} + \epsilon\mathbf{a}$ in which $\mathbf{a}$ is a zero-mean i.i.d random vector of unit variance and in practice ϵ gets small positive values (in theory $\epsilon \longrightarrow 0$).

Performance of the proposed estimator in (22) is evaluated in Figs. 4 and 5. As can be seen in Fig. 4, the optimized filter weights for the labeled pixels are computed based on the estimated iteration and truncation parameters. The actual and estimated MSE plots are shown in Fig. 5 where as can be seen, the estimated shrinkage parameters are very close to their actual values.

III. Filter Approximation

Until now, our statistical analysis was based on the fact that we can compute the filter $\mathbf{W}$ for the whole image. This filter is $n \times n$ for an image containing n pixels, which obviously demands a high computational and storage cost. Since we only need the eigen-decomposition of this matrix, we can approximate the first p eigenvectors and eigenvalues of it without direct computation of all elements of $\mathbf{W}$. Although this idea has not been studied for the purpose of filtering before, [16] used it in the context of spectral grouping, where at first the matrix $\mathbf{K}$ is approximated by means of the Nyström

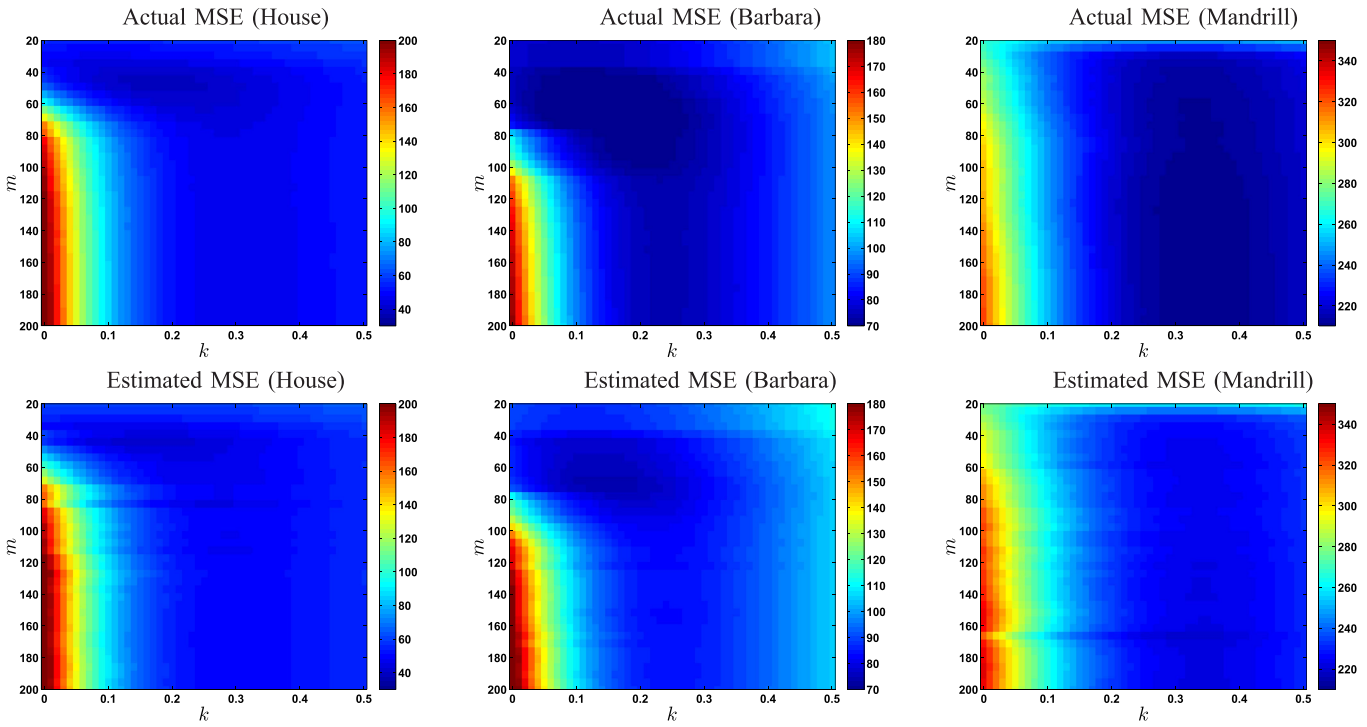


Fig. 5. Corresponding MSE of the images in Fig. 4. The ideal and estimated iteration and truncation numbers are respectively: House (ideal: 0.19, 45, estimated: 0.16, 40), Barbara (ideal: 0.14, 65, estimated: 0.14, 65), Mandrill (ideal: 0.34, 160, estimated: 0.33, 165).

method [13], and then a symmetric normalized version of this matrix is used for a data clustering scheme. Our objective is different because in the end we need to approximate the filtering matrix $\mathbf{W}$, hence we first review what is done in [16] and then adapt it to the approximation we need to effect here.

In the following, the Nyström approach for approximating the similarity (affinity) matrix $\mathbf{K}$ is used first and then the Sinkhorn method (sec. III-B) is applied to estimate the eigen-decomposition of the symmetric, doubly-stochastic filter $\mathbf{W}$. Since the approximated eigenvectors are not exactly orthogonal, finally an orthogonalization procedure is employed to obtain an orthonormal approximation for eigen-decomposition of $\mathbf{W}$. These steps are shown in Algorithm 1 and we will discuss them in more details below.

A. Nyström Approximation

This method is a numerical approximation for estimating the eigenvectors of the symmetric kernel matrix $\mathbf{K}$:

$$\mathbf{K} = \mathbf{\Phi}\mathbf{\Pi}\mathbf{\Phi}^T \tag{23}$$

where $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n]$ represents the orthonormal eigenvectors and $\mathbf{\Pi} = [\pi_1, \pi_2, \ldots, \pi_n]$ contains the eigenvalues of $\mathbf{K}$. Nyström [13] suggests that instead of computing all the entries of $\mathbf{K}$, we can sample our data points and estimate the leading eigenvectors of the matrix $\mathbf{K}$ and, as a result, an approximation $\widetilde{\mathbf{K}}$ can then be built from those estimated eigenvectors.

Having p pixels in a sampled subimage $\mathbf{A}$, we can compute the $p \times p$ kernel matrix $\mathbf{K}_A$ which represents the similarity weights of pixels in $\mathbf{A}$. We also define the subimage $\mathbf{B}$ containing the rest of $(n-p)$ pixels, followed by the $p \times (n-p)$ matrix $\mathbf{K}_{AB}$, which contains the kernel weights between pixels in $\mathbf{A}$ and $\mathbf{B}$. The similarity matrix $\mathbf{K}$ in block form is therefore:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_A & \mathbf{K}_{AB} \\ \mathbf{K}_{AB}^T & \mathbf{K}_B \end{bmatrix} \tag{24}$$

where $\mathbf{K}_B$ denotes the $(n-p) \times (n-p)$ similarity weights between pixels in the subimage[3] $\mathbf{B}$. As can be seen, (24) can be thought of as a permutation of the old $\mathbf{K}$. Nyström suggests the following approximation for the first p eigenvectors of $\mathbf{K}$:

$$\widetilde{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi}_A \\ \mathbf{K}_{AB}^T \mathbf{\Phi}_A \mathbf{\Pi}_A^{-1} \end{bmatrix} \tag{25}$$

where $\mathbf{K}_A = \mathbf{\Phi}_A \mathbf{\Pi}_A \mathbf{\Phi}_A^T$. Intuitively, we can say that the first p entries of $\widetilde{\mathbf{\Phi}}$ are computed exactly, and the $(n-p)$ remaining ones are approximated by a weighted projection of $\mathbf{K}_{AB}$ over the eigenvectors of $\mathbf{K}_A$. Then the approximated similarity matrix will be:

$$\begin{aligned} \widetilde{\mathbf{K}} &= \widetilde{\mathbf{\Phi}} \mathbf{\Pi}_A \widetilde{\mathbf{\Phi}}^T \\ &= \begin{bmatrix} \mathbf{\Phi}_A \\ \mathbf{K}_{AB}^T \mathbf{\Phi}_A \mathbf{\Pi}_A^{-1} \end{bmatrix} \mathbf{\Pi}_A \begin{bmatrix} \mathbf{\Phi}_A^T & \mathbf{\Pi}_A^{-1} \mathbf{\Phi}_A^T \mathbf{K}_{AB} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{K}_A & \mathbf{K}_{AB} \\ \mathbf{K}_{AB}^T & \mathbf{K}_{AB}^T \mathbf{K}_A^{-1} \mathbf{K}_{AB} \end{bmatrix} \end{aligned} \tag{26}$$

Comparing (24) and (26) it can be seen that the huge matrix $\mathbf{K}_B$ is approximated by $\mathbf{K}_{AB}^T \mathbf{K}_A^{-1} \mathbf{K}_{AB}$.

A key aspect of the Nyström approximation is the sampling procedure in which the columns (or rows) of the original $\mathbf{K}$ are selected. The Nyström method was first introduced by a uniform distribution sampling over data [12]. Efficiency of the uniform sampling has been explored in many practical applications [15], [16]. More recently, theoretical aspects

[3] When $n \gg p$, $\mathbf{K}_B$ can be huge.

Algorithm 1 Spectral Approximation of the Filter $\mathbf{W}$

Input: Sub-blocks of the similarity matrix $\mathbf{K}$: $\{\mathbf{K}_A, \mathbf{K}_{AB}\}$, let $(p, p) = \text{size}(\mathbf{K}_A)$ and $(p, n-p) = \text{size}(\mathbf{K}_{AB})$

Output: p leading orthogonal eigenvectors and eigenvalues of the approximated filter $\widetilde{\mathbf{W}}$: $\{\widetilde{\mathbf{V}}, \widetilde{\mathbf{S}}\}$

Nyström Approximation:

1- $\mathbf{K}_A = \mathbf{\Phi}_A \mathbf{\Pi}_A \mathbf{\Phi}_A^T$;

$\Rightarrow$ Eigen-decomposition of the sub-block $\mathbf{K}_A$

2- $\widetilde{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi}_A \\ \mathbf{K}_{AB}^T \mathbf{\Phi}_A \mathbf{\Pi}_A^{-1} \end{bmatrix}$;

$\Rightarrow$ Approximate the m leading eigenvectors of $\mathbf{K}$

*Sinkhorn**:

3- $\mathbf{r} = \text{ones}(n, 1)$; $\boldsymbol{\pi}_A = \text{diag}(\mathbf{\Pi}_A)$;

4- for $i = 1 : iter$

$\mathbf{c} = 1./\left(\widetilde{\mathbf{\Phi}}(\boldsymbol{\pi}_A.(\widetilde{\mathbf{\Phi}}^T \mathbf{r}))\right)$; $\Rightarrow$ Column normalization

$\mathbf{r} = 1./\left(\widetilde{\mathbf{\Phi}}(\boldsymbol{\pi}_A.(\widetilde{\mathbf{\Phi}}^T \mathbf{c}))\right)$; $\Rightarrow$ Row normalization

end

5- for $i = 1 : p$

$\mathbf{W}_{A,AB} = \mathbf{r}(i)(\boldsymbol{\pi}_A^T.\widetilde{\mathbf{\Phi}}(i, :))(\text{repmat}(\mathbf{c}, [1, p]).\widetilde{\mathbf{\Phi}})^T$;

end

6- $\mathbf{W}_A = \mathbf{W}_{A,AB}(:, 1 : p)$;

$\Rightarrow$ Sub-block of the symmetrized $\mathbf{W}_{sym}$

7- $\mathbf{W}_{AB} = \mathbf{W}_{A,AB}(:, p+1 : n)$;

$\Rightarrow$ Sub-block of the symmetrized $\mathbf{W}_{sym}$

Orthogonalization:

8- $\mathbf{W}_A^{1/2} = \text{sqrtm}(\mathbf{W}_A)$;

9- $\mathbf{Q} = \mathbf{W}_A + \mathbf{W}_A^{-1/2} \mathbf{W}_{AB} \mathbf{W}_{AB}^T \mathbf{W}_A^{-1/2}$;

10- $\mathbf{Q} = \mathbf{V}_Q \mathbf{S}_Q \mathbf{V}_Q^T$;

$\Rightarrow$ Eigen-decomposition of the symmetric matrix $\mathbf{Q}$

11- $\widetilde{\mathbf{V}} = \begin{bmatrix} \mathbf{W}_A \\ \mathbf{W}_{AB}^T \end{bmatrix} \mathbf{W}_A^{-1/2} \mathbf{V}_Q \mathbf{S}_Q^{-1/2}$;

$\Rightarrow$ Approximated orthogonal eigenvectors

12- $\widetilde{\mathbf{S}} = \mathbf{S}_Q$; $\Rightarrow$ Approximated eigenvalues

* In the iterative row and column normalization process, construction of the matrix $\widetilde{\mathbf{K}}$ is avoided by using element-wise multiplication.

of nonuniform sampling techniques on real-world data sets have been studied [24], [25]. In general, these nonuniform sampling procedures are biased toward selection of the most informative points of the data-set. However, due to the imposed complexity of the nonuniform distribution updating procedure, practical application of these adaptive methods is limited.

In the current framework, our data are images which contain a high degree of spatial correlation between pixels. This leads us to use *spatially uniform* sampling instead of the random sampling procedure. Spatially uniform sampling is a simple but effective approach in which the spatial distance of the samples are always equally fixed.

To study the performance of the Nyström approximation, we evaluate the relative accuracy defined in [25]:

$$\textit{Relative Accuracy} = \frac{\|\mathbf{K} - \mathbf{K}_{(r)}\|_F}{\|\mathbf{K} - \widetilde{\mathbf{K}}_{(r)}\|_F} \times 100$$

where $\mathbf{K}$ and $\mathbf{K}_{(r)}$ are the actual kernel and its exact rank-r approximation. The approximated kernel $\widetilde{\mathbf{K}}_{(r)}$ is reconstructed by using r leading eigenvectors from the Nyström method. The relative accuracy is lower bounded by zero and will ideally approach 100%.

The relative accuracy of approximating the globalized NLM kernel [3] as a function of the sampling rate is shown for some benchmark images in Fig. 6. We fixed $r = 50$ to capture about 90% of the spectral energy of the global kernel for each image. The samples are uniformly selected over the image lattice, and the relative accuracy is averaged for 20 sampling realizations. It can be seen that while higher sampling percentage leads to smaller error in the approximated kernel matrix, a saturation point is reached beyond 20% sampling density. Furthermore, for a fixed sampling rate the error depends on the contents of the underlying image. Surprisingly, textured images with high frequency components such as Mandrill produce less error compared to smooth images like House. This observation is consistent with results of [26] where it is shown that the error of the Nyström approximation is proportional to coherency of the kernel eigenvectors.

One could assume that at this point we can easily compute our approximated $\mathbf{W}$ and we are done! But as discussed earlier, statistical analysis of this filter needs access to its eigen-decomposition. Constructing a huge $\mathbf{W}$ matrix and then computing its eigenvectors is too expensive. Instead, in the following we explore an efficient way to find the eigenvectors of $\mathbf{W}$.

B. Sinkhorn

The filter $\mathbf{W}$ is the row-normalized kernel matrix $\mathbf{K}$:

$$\mathbf{W} = \mathbf{D}^{-1}\mathbf{K} \tag{27}$$

where $\mathbf{D} = \text{diag}[\sum_{j=1}^n K_{1j}, \sum_{j=1}^n K_{2j}, .., \sum_{j=1}^n K_{nj}]$. We approximate the matrix $\mathbf{W}$ with a doubly-stochastic (symmetric) positive definite matrix, using Sinkhorn's algorithm [20]. Based on this method, given a positive valued matrix $\mathbf{K}$, there exist diagonal matrices $\mathbf{R} = \text{diag}(\mathbf{r})$ and $\mathbf{C} = \text{diag}(\mathbf{c})$ such that $\mathbf{W}_{sym} = \mathbf{RKC}$.

Since we have estimated the leading eigenvectors of $\mathbf{K}$, there is no need to compute $\mathbf{RKC}$. Instead, as can be seen in Algorithm. 1, $\mathbf{W}_{sym}$ is approximated by its two sub-blocks $\mathbf{W}_A$ and $\mathbf{W}_{AB}$ where:

$$\mathbf{W}_{sym} = \begin{bmatrix} \mathbf{W}_A & \mathbf{W}_{AB} \\ \mathbf{W}_{AB}^T & \mathbf{W}_B \end{bmatrix} \tag{28}$$

Again, the Nyström method could give the approximated eigenvectors, but the only minor problem is that these eigenvectors are not quite orthogonal. In the following we discuss an approximation of the orthogonal eigenvectors.

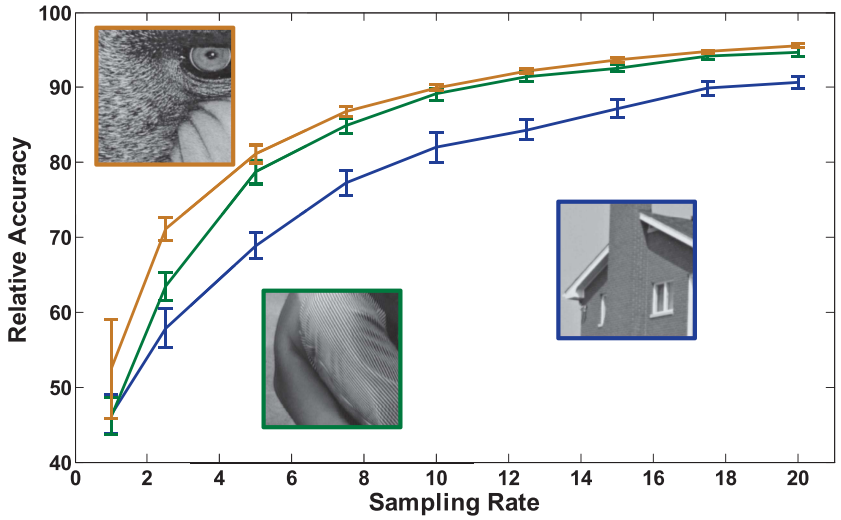


Fig. 6. Accuracy of the kernel approximation for different sampling rates (sampling rate percentage is defined as $\frac{p}{n} \times 100\%$ where p denotes the number of samples and n represents number of pixels in the image). For the ease of computation of the exact filter, 150×150 subimages of Mandrill, Barbara and House are selected.

C. Orthogonalization

With the two sub-blocks $\mathbf{W}_A$ and $\mathbf{W}_{AB}$ in hand, here we derive an expression for approximating the orthogonalized eigenvectors $\widetilde{\mathbf{V}}$. As discussed in [16], for any positive definite matrix, the orthogonalized approximated eigenvectors can be solved in one step. Let $\mathbf{W}_A^{1/2}$ denote the symmetric positive definite square root of $\mathbf{W}_A$. We define $\mathbf{Q} = \mathbf{W}_A + \mathbf{W}_A^{-1/2}\mathbf{W}_{AB}\mathbf{W}_{AB}^T\mathbf{W}_A^{-1/2}$ and we also consider the eigen-decomposition of this symmetric matrix as $\mathbf{Q} = \mathbf{V}_Q\mathbf{S}_Q\mathbf{V}_Q^T$. Then, it can been shown that the approximated symmetric $\widetilde{\mathbf{W}}$ is diagonalized by $\widetilde{\mathbf{S}} = \mathbf{S}_Q$ and $\widetilde{\mathbf{V}}$ where:

$$\widetilde{\mathbf{V}} = \begin{bmatrix} \mathbf{W}_A \\ \mathbf{W}_{AB}^T \end{bmatrix} \mathbf{W}_A^{-1/2}\mathbf{V}_Q\mathbf{S}_Q^{-1/2} \tag{29}$$

Then the approximated filter can be expressed as:

$$\widetilde{\mathbf{W}} = \widetilde{\mathbf{V}}\widetilde{\mathbf{S}}\widetilde{\mathbf{V}}^T \tag{30}$$

Proof of this approximation is given in Appendix C.

The described three-step procedure provides us with an approximation of the leading eigenvectors and eigenvalues of the filter. Denoising performance of the approximated filter is compared to the exact filter in Fig. 7. These results suggest that the proposed approximation with a small sampling rate can almost reach performance of the exact filter.

IV. Experiments

In this section performance of our algorithm is compared to state-of-the-art denoising methods for some benchmark images. We selected NLM [3] as our baseline kernel; however, any other non-local kernel could also be used. Pixel samples of the Nyström extension are uniformly selected and the sampling rate is set as 1% ($p = \frac{n}{100}$) and is kept fixed throughout the experiments.

Performance of the proposed filter is quantified across different noise levels in Table I. For each noise level, we report the Monte-Carlo average performance for each algorithm over 5 different noise realizations. We highlight (in bold) both the best results, and also results that are statistically within the margin of standard error from the best results (0.05 dB in PSNR and 0.005 SSIM). In this set of experiments, the pre-filtered images are obtained from NLM. As can be seen, our method consistently improves upon NLM in terms of PSNR and SSIM index [27], especially for high noise levels where local similar pixels are more difficult to find.

Fig. 8 demonstrates the denoising results obtained by NLM compared to the proposed method. In addition to the PSNR improvement, visual quality of the proposed method also is superior to the NLM filter. As it can be seen, both edges and smooth features of the image are preserved better than the other methods.

In the next set of experiments shown in Table II, the pre-filtered images are obtained from BM3D [4]. Our method can improve upon BM3D especially at high noise levels and for images with semi-stochastic textures which contain relatively few similar patches.

Denoising results of the Mandrill and Monarch images for BM3D and the globalized BM3D are compared in Fig. 9. As can be seen, the proposed method can bootstrap the performance of BM3D.

Since in practice the distribution of the noise is not additive white Gaussian, we also tested our algorithm for real noise in color images.[4] In this set of experiments the best results are optimized using the no-reference quality metric in [28]. Fig. 10 shows performance of the proposed method for improving the NLM filter. We also compare our results to the commercial Neat Image™denoising software in Fig. 11. As can be seen, our result is competitive to the commercial state-of-the art denoising. We note that our Matlab code and additional results are available at the project website.[5]

Running time for denoising a 256×256 grayscale image with an unoptimized implementation of our method is

[4]The color denoising is applied in YUV space, where the weights are computed from the Y channel, and applied to the U and V channels.

[5]http://www.soe.ucsc.edu/~htalebi/GLIDE.php.

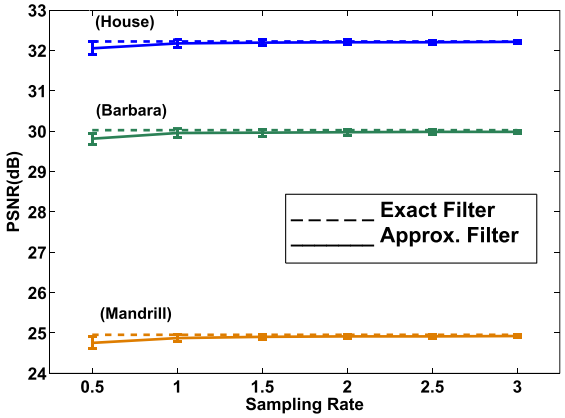


Fig. 7. Comparison of the denoising performance (AWG with $\sigma = 20$) of the exact and approximated filter for the subimages in Fig. 6.

TABLE I

PSNR VALUES OF NLM [3] (1ST COLUMN), AND THE PROPOSED METHOD (2ND COLUMN). RESULTS NOTED ARE AVERAGE PSNR (TOP) AND SSIM [27] (BOTTOM) OVER 5 INDEPENDENT NOISE REALIZATIONS FOR EACH σ

σ	Monarch		House		Cameraman		Aerial		Mandrill		Stream		Avgerage Improvement
	NLM	G-NLM	NLM	G-NLM	NLM	G-NLM	NLM	G-NLM	NLM	G-NLM	NLM	G-NLM	
10	33.43	**33.50**	35.25	**35.46**	33.43	**33.62**	30.54	**30.78**	30.01	**30.30**	30.54	**30.79**	0.22
	0.940	**0.945**	0.893	**0.901**	0.914	**0.921**	**0.930**	**0.932**	0.891	**0.903**	0.893	**0.903**	0.009
20	29.26	**29.67**	32.37	**32.63**	29.41	**29.49**	26.23	**26.59**	25.69	**25.96**	26.36	**26.67**	0.29
	0.894	**0.903**	0.847	**0.857**	0.834	**0.852**	0.829	**0.834**	0.769	**0.781**	0.736	**0.748**	0.014
30	27.16	**27.65**	30.18	**30.57**	27.54	**27.78**	24.36	**24.65**	23.85	**24.42**	24.69	**25.19**	0.39
	0.841	**0.863**	0.796	**0.827**	0.786	**0.817**	0.755	**0.770**	0.652	**0.684**	0.646	**0.671**	0.031
40	25.53	**26.32**	28.32	**29.01**	25.98	**26.47**	23.01	**23.34**	22.69	**22.36**	23.61	**24.29**	0.62
	0.778	**0.833**	0.721	**0.799**	0.715	**0.787**	0.691	**0.713**	0.578	**0.628**	0.579	**0.630**	0.055
50	24.44	**25.24**	26.94	**27.73**	24.90	**25.28**	21.95	**22.39**	21.80	**22.64**	22.75	**23.56**	0.71
	0.727	**0.802**	0.670	**0.773**	0.657	**0.735**	0.629	**0.658**	0.507	**0.584**	0.518	**0.592**	0.074

TABLE II

PSNR VALUES OF BM3D [4] (1ST COLUMN), AND THE PROPOSED METHOD (2ND COLUMN). RESULTS NOTED ARE AVERAGE PSNR (TOP) AND SSIM [27] (BOTTOM) OVER 5 INDEPENDENT NOISE REALIZATIONS FOR EACH σ

σ	Monarch		House		Cameraman		Aerial		Mandrill		Stream		Average Improvement
	BM3D	G-BM3D	BM3D	G-BM3D	BM3D	G-BM3D	BM3D	G-BM3D	BM3D	G-BM3D	BM3D	G-BM3D	
10	**34.12**	34.08	**36.67**	36.70	**34.05**	34.06	**31.09**	31.07	30.57	**30.65**	**31.14**	31.17	0.02
	0.956	**0.956**	**0.920**	**0.921**	**0.930**	**0.932**	**0.938**	**0.939**	0.896	**0.902**	**0.906**	**0.909**	0.002
20	30.42	**30.56**	**33.78**	**33.81**	**30.41**	**30.45**	27.22	**27.32**	26.58	**26.71**	27.25	**27.33**	0.08
	0.920	**0.923**	0.871	**0.872**	0.874	**0.878**	0.864	**0.869**	0.790	**0.802**	0.790	**0.798**	0.006
30	28.42	**28.52**	**32.04**	**32.09**	**28.58**	**28.61**	25.24	**25.31**	24.53	**24.70**	25.46	**25.59**	0.09
	0.885	**0.889**	0.846	**0.849**	0.835	**0.840**	0.798	**0.806**	0.697	**0.728**	0.700	**0.719**	0.012
40	26.66	**26.85**	30.56	**30.61**	27.09	**27.20**	23.77	**23.90**	23.07	**23.25**	24.32	**24.51**	0.15
	0.846	**0.851**	0.822	**0.828**	0.804	**0.819**	0.737	**0.740**	0.614	**0.652**	0.630	**0.654**	0.015
50	25.72	**25.98**	29.64	**29.73**	26.05	**26.28**	22.94	**23.14**	22.32	**22.57**	23.57	**23.79**	0.21
	0.820	**0.827**	0.809	**0.813**	0.779	**0.801**	0.690	**0.705**	0.545	**0.587**	0.575	**0.596**	0.019

about 160 seconds on a 2.8 GHz Intel Core i7 processor. However, parallelizing can significantly speed up our method. For instance, running time of the parallelized version of our code executed with 4 separate cores takes about 50 seconds.

V. CONCLUSION

This work is, to our knowledge, the very first truly global denoising algorithm to be proposed. The global approach goes beyond the dominant paradigm of non-local patch-based processing, which we have shown here to be inherently limited. The specific contribution we have made is to develop a practical algorithm to compute a global filter which in effect uses all the pixels in the input image to denoise every single pixel. By exploiting the Nyström extension, we have made the global approach computationally tractable. Since the global filter uses all the pixels of the image, exact computation of the filter weights has a complexity $O(n^2)$, whereas the proposed sample based approximation, the complexity is reduced to linear time $O(pn)$, where p is the number of samples, typically a small fraction of the total number of pixels. At the same time, the experimental results demonstrated that the proposed approach improves over the best existing patch-based methods in terms of both PSNR and subjective visual quality. While this improvement is modest, it is only a starting point, as we have good reason to believe that the improvement in performance brought by the global approach will grow substantially with increasing image size. In an upcoming work, we will present a more detailed analysis of the asymptotic performance of global denoising filters and quantify this gain as a function of image size and the degrees of freedom implied by the the image content.

To better understanding the global filter in this paper, we studied the oracle performance of the proposed method and compared this to the oracle performance of other (mainly

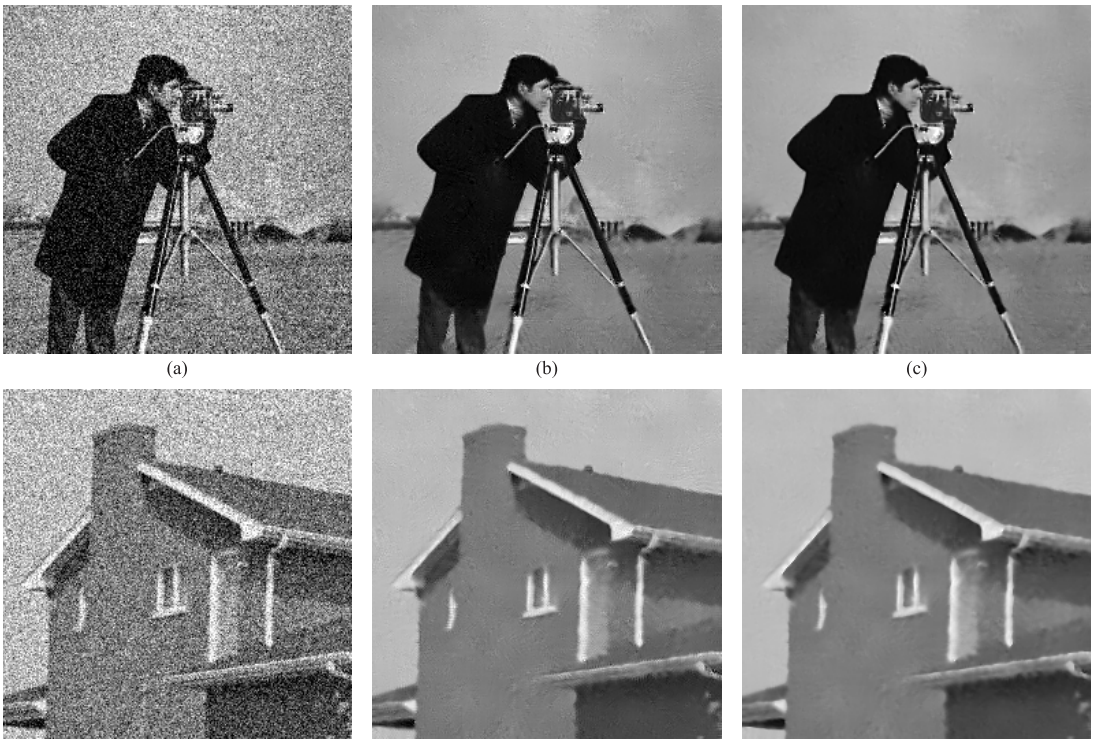


Fig. 8. Comparison of denoising performance on noisy images corrupted by AWGN of $\sigma = 40$. (a), (d) Noisy input, (b), (e) NLM [3], (c), (f) G-NLM.

patch-based) methods in Table III.[6] As can be seen, the oracle GLIDE outperforms other oracle methods by a significant margin. While this margin is only a bound on how much improvement we can expect in practice, it does convey an interesting and tantalizing message. Namely (at least asymptotically) patch-based methods are inherently limited in performance [9] in a way that global filtering is not. More specifically, the oracle PSNR values for the global filter point to essentially perfect reconstruction of the noise-free image, which is apparently impossible to achieve for oracle versions of algorithms like such as BM3D, even if all the filter parameters are known exactly.

APPENDIX A
MSE OF THE WHOLE IMAGE

From (8) we can show that each row of $\mathbf{W}$ can be expressed as:

$$\mathbf{w}_i^T = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j(i)\mathbf{v}_j^T, \tag{31}$$

where $\mathbf{v}_j(i)$ denotes the i-th entry of the j-th eigenvector. Then each estimated pixel $\widehat{z}_i$ has the following form:

$$\widehat{z}_i = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j(i)\mathbf{v}_j^T \mathbf{y}, \tag{32}$$

[6]The oracle NLM has all the kernel weights computed from the clean image and in the case of the oracle BM3D [4] which has been shown to be near optimal [29] (among patch-based methods), the pre-filtered image is replaced by the clean image (which means that the Wiener shrinkage and the patch grouping are implemented perfectly). Similarly, the oracle GLIDE has all the global weights computed from the clean image.

Bias of this estimate can be expressed as:

$$\begin{aligned}\text{bias}(\widehat{z}_i) &= z_i - \text{E}(\widehat{z}_i) = \sum_{j=1}^{n} \mathbf{v}_j(i)\mathbf{v}_j^T\mathbf{z} - \sum_{j=1}^{n} \lambda_j \mathbf{v}_j(i)\mathbf{v}_j^T\mathbf{z} \\ &= \sum_{j=1}^{n} (1-\lambda_j)\mathbf{v}_j(i)b_j \end{aligned} \tag{33}$$

where $\mathbf{b} = \mathbf{V}^T\mathbf{z} = [b_1, \ldots, b_n]^T$ contains the projected signal in all modes. The variance term also has the following form:

$$\begin{aligned}\text{var}(\widehat{z}_i) &= \sigma^2(\mathbf{w}_i^T\mathbf{w}_i) = \sigma^2(\sum_{j=1}^{n} \lambda_j \mathbf{v}_j(i)\mathbf{v}_j^T)(\sum_{j'=1}^{n} \lambda_{j'}\mathbf{v}_{j'}(i)\mathbf{v}_{j'}) \\ &= \sigma^2 \sum_{j=1}^{n} \lambda_j^2 \mathbf{v}_j(i)^2 \end{aligned} \tag{34}$$

where in the last equation we have $\mathbf{v}_j^T\mathbf{v}_{j'} = \delta_{jj'}$ for the orthonormal basis functions. Overall, the MSE of the i-th estimated pixel is:

$$\begin{aligned}\text{MSE}_i &= \text{bias}(\widehat{z}_i)^2 + \text{var}(\widehat{z}_i) \\ &= (\sum_{j=1}^{n} (1-\lambda_j)\mathbf{v}_j(i)b_j)^2 + \sigma^2 \sum_{j=1}^{n} \lambda_j^2 \mathbf{v}_j(i)^2. \end{aligned} \tag{35}$$

This expression can be used to analyze the framework given in (32).

The estimated MSE of the whole image is given by:

$$\text{MSE} = \sum_{i=1}^{n} \text{MSE}_i = \sum_{i=1}^{n} \text{bias}(\widehat{z}_i)^2 + \sum_{i=1}^{n} \text{var}(\widehat{z}_i) \tag{36}$$

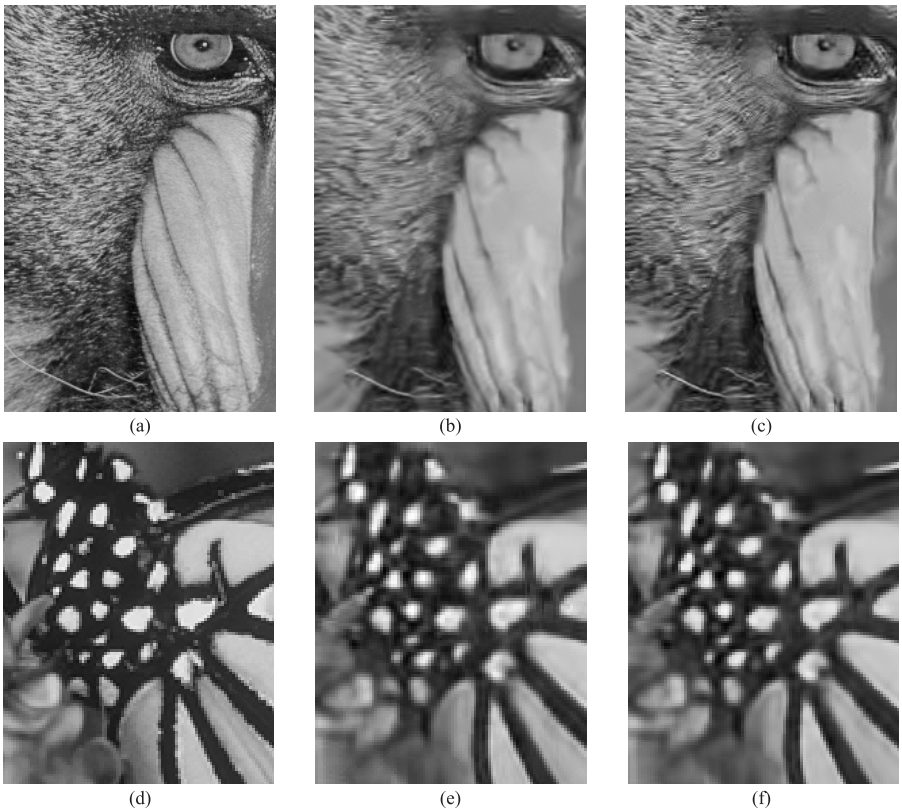


Fig. 9. Comparison of denoising performance on noisy images corrupted by AWGN of $\sigma = 50$. (a), (d) Original image, (b), (e) BM3D [4], (c), (f) G-BM3D.

Reminding the orthonormality of the eigenvectors $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$, the variance term can be written as:

$$\sum_{i=1}^{n} \mathrm{var}(\widehat{z}_i) = \sum_{i=1}^{n} \sigma^2 \sum_{j=1}^{n} \lambda_j^2 \mathbf{v}_j(i)^2 = \sigma^2 \sum_{j=1}^{n} \lambda_j^2 \sum_{i=1}^{n} \mathbf{v}_j(i)^2 = \sigma^2 \sum_{j=1}^{n} \lambda_j^2 \quad (37)$$

where the last equation comes from $\sum_{i=1}^{n} \mathbf{v}_j(i)^2 = 1$. We also can write the bias term as follows:

$$\begin{aligned}\sum_{i=1}^{n} \mathrm{bias}(\widehat{z}_i)^2 &= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} (1-\lambda_j)\mathbf{v}_j(i) b_j \right)^2 \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} ((1-\lambda_j)^2 \mathbf{v}_j(i)^2 b_j^2 \\ &\quad + (1-\lambda_j)\mathbf{v}_j(i) b_j \sum_{l \neq j}^{n} (1-\lambda_l)\mathbf{v}_l(i) b_l) \\ &= \sum_{j=1}^{n} ((1-\lambda_j)^2 b_j^2 \sum_{i=1}^{n} \mathbf{v}_j(i)^2 \\ &\quad + (1-\lambda_j) b_j \sum_{l \neq j}^{n} (1-\lambda_l) b_l \sum_{i=1}^{n} \mathbf{v}_l(i)\mathbf{v}_j(i)) \\ &= \sum_{j=1}^{n} (1-\lambda_j)^2 b_j^2 \end{aligned} \quad (38)$$

where in the last equation $\sum_{i=1}^{n} \mathbf{v}_l(i)\mathbf{v}_j(i) = 0$ with $l \neq j$. From what we have for the squared bias and variance we can conclude:

$$\mathrm{MSE} = \sum_{j=1}^{n} (1-\lambda_j)^2 b_j^2 + \sigma^2 \lambda_j^2. \quad (39)$$

Appendix B

MSE Analysis of the Truncated Filter

Each row of the truncated filter can be expressed as:

$$\widetilde{\mathbf{w}}_i^T = \sum_{j=1}^{m} \lambda_j \mathbf{v}_j(i)\mathbf{v}_j^T, \quad (40)$$

where $\mathbf{v}_j(i)$ denotes the i-th entry of the j-th eigenvector. Then each estimated pixel $\widehat{z}_i$ has the following form:

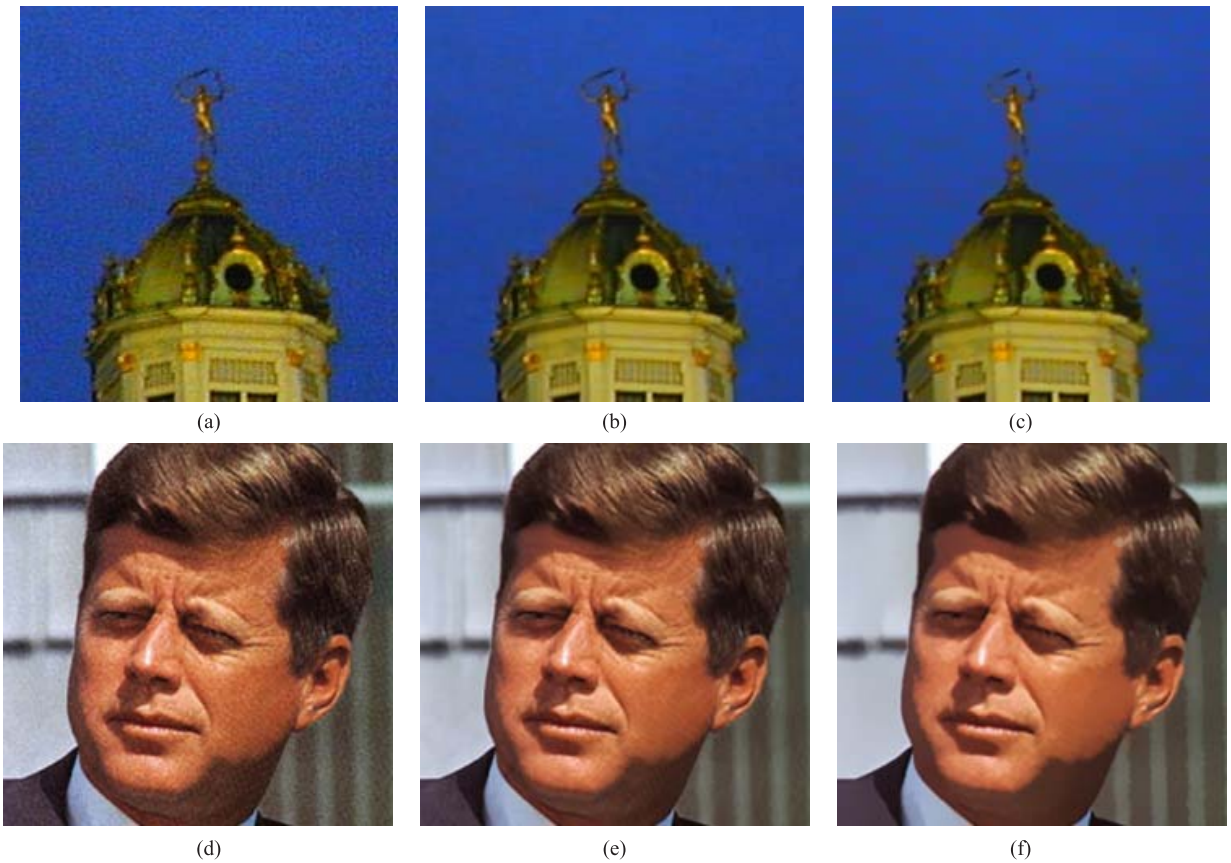
$$\widehat{z}_i = \sum_{j=1}^{m} \lambda_j \mathbf{v}_j(i)\mathbf{v}_j^T \mathbf{y}, \quad (41)$$

Bias of this estimate can be expressed as:[7]

$$\mathrm{bias}(\widehat{z}_i) = z_i - \mathrm{E}(\widehat{z}_i) = z_i - \sum_{j=1}^{m} \lambda_j \mathbf{v}_j(i) b_j \quad (42)$$

where $\mathbf{b} = \mathbf{V}^T\mathbf{z} = [b_1, \ldots, b_m]^T$ contains the projected signal in the first m modes. The variance term also has the following

[7]It is worth pointing out that in the truncated space $z_i \neq \sum_{j=1}^{m} \mathbf{v}_j(i)\mathbf{v}_j^T\mathbf{z}$, because $\mathbf{V}\mathbf{V}^T \neq \mathbf{I}$.



(a) (b) (c) (d) (e) (f)

Fig. 10. Comparison of denoising performance on the real noise. (a) and (d) Noisy image, (b) and (e) CBM3D [4], (c) and (f) G-NLM.

form:

$$\begin{aligned}\mathrm{var}(\widehat{z}_i) &= \sigma^2(\widetilde{\mathbf{w}}_i^T\widetilde{\mathbf{w}}_i) = \sigma^2(\sum_{j=1}^{m}\lambda_j\mathbf{v}_j(i)\mathbf{v}_j^T)(\sum_{j'=1}^{m}\lambda_{j'}\mathbf{v}_{j'}(i)\mathbf{v}_{j'}) \\ &= \sigma^2\sum_{j=1}^{m}\lambda_j^2\mathbf{v}_j(i)^2 \end{aligned} \tag{43}$$

The MSE of the i-th estimated pixel is:

$$\begin{aligned}\mathrm{MSE}_i &= \mathrm{bias}(\widehat{z}_i)^2 + \mathrm{var}(\widehat{z}_i) \\ &= (z_i - \sum_{j=1}^{m}\lambda_j\mathbf{v}_j(i)b_j)^2 + \sigma^2\sum_{j=1}^{m}\lambda_j^2\mathbf{v}_j(i)^2 \\ &= z_i^2 + \sum_{j=1}^{m}(\lambda_j^2(b_j^2+\sigma^2)\mathbf{v}_j(i)^2 \\ &\quad - 2z_i\lambda_j\mathbf{v}_j(i)b_j) \end{aligned} \tag{44}$$

Having MSE_i, we can show that the total MSE for the whole image is:

$$\mathrm{MSE} = \sum_{i=1}^{n}\mathrm{MSE}_i = \sum_{i=1}^{n}z_i^2 + \sum_{j=1}^{m}\left((\lambda_j^2 - 2\lambda_j)b_j^2 + \sigma^2\lambda_j^2\right) \tag{45}$$

where $\|\mathrm{bias}(\widehat{\mathbf{z}})\|^2 = \sum_{i=1}^{n}z_i^2 + \sum_{j=1}^{m}(\lambda_j^2 - 2\lambda_j)b_j^2$ and $\mathrm{var}(\widehat{\mathbf{z}}) = \sigma^2\sum_{j=1}^{m}\lambda_j^2$.

APPENDIX C
EIGENVECTOR ORTHOGONALIZATION

Having the two sub-blocks $\mathbf{W}_A$ and $\mathbf{W}_{AB}$ of the filter $\mathbf{W}$ and defining $\mathbf{Q} = \mathbf{W}_A + \mathbf{W}_A^{-1/2}\mathbf{W}_{AB}\mathbf{W}_{AB}^T\mathbf{W}_A^{-1/2}$ with the eigen-decomposition $\mathbf{Q} = \mathbf{V}_Q\mathbf{S}_Q\mathbf{V}_Q^T$, we aim to show that the orthonormal eigenvector bases for the estimated filter $\widetilde{\mathbf{W}}$ are:

$$\widetilde{\mathbf{V}} = \begin{bmatrix}\mathbf{W}_A \\ \mathbf{W}_{AB}^T\end{bmatrix}\mathbf{W}_A^{-1/2}\mathbf{V}_Q\mathbf{S}_Q^{-1/2} \tag{46}$$

We first need to check $\widetilde{\mathbf{W}} = \widetilde{\mathbf{V}}\mathbf{S}_Q\widetilde{\mathbf{V}}^T$:

$$\begin{aligned}\widetilde{\mathbf{W}} &= \left\{\begin{bmatrix}\mathbf{W}_A \\ \mathbf{W}_{AB}^T\end{bmatrix}\mathbf{W}_A^{-1/2}\mathbf{V}_Q\mathbf{S}_Q^{-1/2}\right\} \\ &\quad \mathbf{S}_Q\{\mathbf{S}_Q^{-1/2}\mathbf{V}_Q^T\mathbf{W}_A^{-1/2}\begin{bmatrix}\mathbf{W}_A & \mathbf{W}_{AB}\end{bmatrix}\} \\ &= \widetilde{\mathbf{V}}\mathbf{S}_Q\widetilde{\mathbf{V}}^T \end{aligned} \tag{47}$$

In addition, we check the orthogonality of $\widetilde{\mathbf{V}}$ as follows:

$$\begin{aligned}\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}} &= \mathbf{S}_Q^{-1/2}\mathbf{V}_Q^T\mathbf{W}_A^{-1/2}\begin{bmatrix}\mathbf{W}_A & \mathbf{W}_{AB}\end{bmatrix} \\ &\quad \begin{bmatrix}\mathbf{W}_A \\ \mathbf{W}_{AB}^T\end{bmatrix}\mathbf{W}_A^{-1/2}\mathbf{V}_Q\mathbf{S}_Q^{-1/2} \\ &= \mathbf{S}_Q^{-1/2}\mathbf{V}_Q^T\mathbf{Q}\mathbf{V}_Q\mathbf{S}_Q^{-1/2} \\ &= \mathbf{I} \end{aligned} \tag{48}$$

As a result, the approximated eigen-decomposition is orthogonal.

(a) (b) (c)

Fig. 11. Comparison of denoising performance on the real noise. (a) Noisy image, (b) Neat Image™, (c) G-NLM. (Neat Image™denoising software is available at http://www.neatimage.com.)

TABLE III

PSNR VALUES OF ORACLE NLM [3] (1ST COLUMN), ORACLE BM3D [4] (2ND COLUMN), AND THE ORACLE GLIDE (3RD COLUMN). RESULTS NOTED ARE AVERAGE PSNR OVER 5 INDEPENDENT NOISE REALIZATIONS FOR EACH σ

σ	Monarch			House			Cameraman			Aerial			Average		
	NLM	BM3D	GLIDE	NLM	BM3D	GLIDE	NLM	BM3D	GLIDE	NLM	BM3D	GLIDE	NLM	BM3D	GLIDE
10	35.01	36.55	**43.16**	36.52	39.29	**51.24**	34.82	36.72	**41.58**	31.95	33.56	**48.75**	34.58	36.53	**46.18**
20	30.87	33.02	**41.83**	34.05	36.54	**47.18**	31.01	33.15	**40.49**	27.78	29.75	**45.37**	30.93	33.12	**43.72**
30	28.95	31.08	**40.41**	32.48	34.97	**44.89**	28.91	31.20	**39.28**	26.13	27.74	**42.99**	29.12	31.25	**41.89**
40	27.62	29.75	**39.27**	31.18	33.82	**43.08**	27.39	29.89	**38.17**	24.91	26.41	**41.21**	27.78	29.97	**40.43**
50	26.84	28.12	**38.25**	29.96	32.48	**41.60**	26.85	28.56	**37.16**	23.85	25.50	**39.86**	26.87	28.67	**39.22**

REFERENCES

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th ICCV*, Jan. 1998, pp. 836–846.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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, Jun. 2013.

[7] 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, pp. 1531–1549, Apr. 2010.

[8] 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.

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

[10] A. Levin, B. Nadler, F. Durand, and W. T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *Proc. ECCV*, Oct. 2012, pp. 73–86.

[11] H. Talebi, X. Zhu, and P. Milanfar, "How to SAIF-ly boost denoising performance," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1470–1485, Apr. 2013.

[12] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Adv. NIPS*, 2001, pp. 682–688.

[13] E. Nyström, "Über die Praktische Auflösung von linearn ingtegraglechungen mit anwendungen auf Randwertaufgaben der potentialtheorie," *Commentationes Phys. Math.*, vol. 4, no. 15, pp. 1–52, Apr. 1928.

[14] C. T. Baker, *The Numerical Treatment of Integral Equations*. Oxford, U.K.: Clarendon Press, 1977.

[15] A. Talwalkar, S. Kumar, and H. Rowley, "Large-scale manifold learning," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[16] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.

[17] Z. Farbman, R. Fattal, and D. Lischinski, "Diffusion maps for edge-aware image editing," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 1–145, 2010.

[18] E. Seneta, *Non-negative Matrices and Markov Chains*. New York, NY, USA: Springer-Verlag, 1981.

[19] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[20] P. Milanfar, "Symmetrizing smoothing filters," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 263–284, 2012.

[21] C. Stein, "Estimation of the mean of a multivariate normal distribution," *Anna. Statist.*, vol. 9, no. 6, pp. 1135–1151, 1981.

[22] G. Wahba, "The fast Monte-Carlo cross-validation and C_L procedures: Comments, new results and applications to image recovery problems-comments," *Comput. Stat.*, vol. 10, no. 3, pp. 249–250, 1995.

[23] S. Ramani, T. Blu, and M. Unser, "Monte-Carlo sure: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008.

[24] P. Drineas and M. W. Mahoney, "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *J. Mach. Learn. Res.*, vol. 6, pp. 2153–2175, Dec. 2005.

[25] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling methods for the Nyström method," *J. Mach. Learn. Res.*, vol. 13, pp. 981–1006, Apr. 2012.

[26] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[27] 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.

[28] 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.

[29] 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.

Hossein Talebi (S'11) received the B.S. and M.S. degrees in electrical engineering from the Isfahan University of Technology, Isfahan, Iran, in 2007 and 2010, respectively. Currently, he is pursuing the Ph.D. degree in electrical engineering with the University of California, Santa Cruz, where he is a Researcher with the Multi-Dimensional Signal Processing Laboratory. His research interests are in image and video processing (i.e., denoising, super-resolution, and editing).

Peyman Milanfar (F'10) is a Professor of electrical engineering with UC Santa Cruz (UCSC), and was an Associate Dean of research from 2010 to 2012. He is currently on leave at Google-[x]. He received the B.S. degree in electrical engineering/mathematics from Berkeley and the Ph.D. degree in electrical engineering and computer science from MIT. Prior to UCSC, he was with SRI, and a Consulting Professor of computer science with Stanford. He founded MotionDSP, which has brought state-of-art video enhancement to market. His technical expertise are in statistical signal, image and video processing, computational photography, and vision. He is a member of the IEEE Signal Processing Society's IVMSP Technical Committee, and its Awards Board.