A General Framework for Regularized, Similarity-Based Image Restoration

Amin Kheradmand, *Student Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

***Abstract*— Any image can be represented as a function defined on a weighted graph, in which the underlying structure of the image is encoded in kernel similarity and associated Laplacian matrices. In this paper, we develop an iterative graph-based framework for image restoration based on a new definition of the normalized graph Laplacian. We propose a cost function, which consists of a new data fidelity term and regularization term derived from the specific definition of the normalized graph Laplacian. The normalizing coefficients used in the definition of the Laplacian and associated regularization term are obtained using fast symmetry preserving matrix balancing. This results in some desired spectral properties for the normalized Laplacian such as being symmetric, positive semidefinite, and returning zero vector when applied to a constant image. Our algorithm comprises of outer and inner iterations, where in each outer iteration, the similarity weights are recomputed using the previous estimate and the updated objective function is minimized using inner conjugate gradient iterations. This procedure improves the performance of the algorithm for image deblurring, where we do not have access to a good initial estimate of the underlying image. In addition, the specific form of the cost function allows us to render the spectral analysis for the solutions of the corresponding linear equations. In addition, the proposed approach is general in the sense that we have shown its effectiveness for different restoration problems, including deblurring, denoising, and sharpening. Experimental results verify the effectiveness of the proposed algorithm on both synthetic and real examples.**

***Index Terms*— Deblurring, kernel similarity matrix, sharpening, graph Laplacian, denoising.**

I. Introduction

MOST real pictures exhibit some amount of degradation depending on the camera and settings used to capture the scene, environmental conditions, and the amount of relative motion between camera and subject, among other factors. Restoration algorithms aim to undo undesired distortions like blur and/or noise from the degraded image. In this paper, we concentrate on problems where the main distortion of the image comes from blurring. We assume linear shift invariant point spread functions (PSFs), such that the blurring process is described through the following linear model

$$\mathbf{y} = A\mathbf{z} + \mathbf{n}. \tag{1}$$

In this model, $\mathbf{y}$ is a lexicographically ordered vector representation of the input $n \times n$ blurred and noisy image, $\mathbf{z}$ is the latent image in vector form, and $\mathbf{n}$ is a noise vector consisting of independent and identically distributed zero mean noise with standard deviation σ. Also, A is an $n^2 \times n^2$ blurring matrix which is constructed from the corresponding PSF and usually has a special structure depending on the type of boundary condition assumptions [1], [2].

Most existing deblurring methods rely on optimizing a cost function of the form

$$E(\mathbf{z}) = \|\mathbf{y} - A\mathbf{z}\|^2 + \eta\ R(\mathbf{z}) \tag{2}$$

with respect to the unknown image vector $\mathbf{z}$. The first term in the above is the data fidelity term and the second term implies a prior term which regularizes the inherently ill-posed problem. In such algorithms, the parameter η controls the amount of regularization to keep the final estimate from being too smooth or exhibiting unpleasant noise amplification and ringing artifacts. Deblurring algorithms can be classified based on the type of blurs they deal with, and also different choices of the regularization term they exploit to solve the deblurring problem [3], [4]. A large class of deblurring algorithms take advantage of a total variation (TV)-type regularization term [5]–[7]. They mostly differ in the specific definition of the TV term and the optimization method for solving the resulting cost function. Other methods use a nonlocal differential operator as the regularization term with different norms [8]–[10]. Sparsity-based methods are also motivated by sparse representation of images in some appropriate domain [11], [12]. In [13], a Hessian norm regularization is used for deblurring, with biomedical applications. Example-based manifold priors are used in [14] to regularize the deblurring problem. In [15], a prior term is added to encourage the estimate to have a gradient distribution similar to a reference. Furthermore, some recent algorithms are based on the idea of decoupling deblurring and denoising and exploiting the powerful BM3D algorithm [16] in their denoising phase [17], [18]. In [18], BM3D frames are defined explicitly and based on a generalized Nash equilibrium approach, the two objective functions for denoising and deblurring parts are balanced. This algorithm is one of the best existing deblurring methods for symmetric blurs (e.g., Gaussian and out-of-focus blurs). For motion deblurring applications [19], [20], a hyper-Laplacian prior

Manuscript received January 27, 2014; revised September 22, 2014; accepted September 22, 2014. Date of publication October 8, 2014; date of current version October 23, 2014. 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 Dr. Brendt Wohlberg.

The authors are with the Department of Electrical Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: aminkh@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.2014.2362059

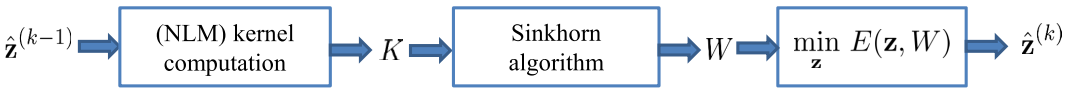


Fig. 1. Block diagram of our proposed iterative deblurring method. K is the kernel similarity matrix, and W is the doubly stochastic filtering matrix. $\hat{\mathbf{z}}^{(k)}$ is the estimate corresponding to optimization of the objective function at the kth outer iteration of the algorithm.

Algorithm 1 Iterative Restoration Algorithm

Inputs: blurred noisy image $\mathbf{y}$, blurring matrix A
Output: deblurred estimate $\mathbf{z}^\star$
Initializations:
1. Estimate the noise standard deviation $\hat{\sigma}$ in $\mathbf{y}$ using algorithm in [43].
2. Denoise $\mathbf{y}$ using denoising algorithm in [16] to derive $\hat{\mathbf{z}}^{(0)}$.
3. Set $k = 0$.
while not converged **do**
- Compute K from $\hat{\mathbf{z}}^{(k)}$ using Eq. (11).
- Apply Sinkhorn algorithm in [30] to K to get the diagonal matrix C.
- Compute the filtering matrix as $W = C^{-1/2}KC^{-1/2}$.
- Solve objective function in (16) using CG to compute $\hat{\mathbf{z}}^{(k+1)}$.
- Set $\mathbf{z}^\star = \hat{\mathbf{z}}^{(k+1)}$, and $k = k + 1$.
end while
return $\mathbf{z}^\star$

based on the statistics of natural images is used. A progressive intra-scale, inter-scale approach is used in [21] for non-blind image deconvolution. Shan et al. [22] have proposed a cost function in which the data fidelity term involves different derivative terms for motion deblurring of natural images.

In this paper, we propose a new approach for kernel similarity-based image deblurring by introducing a novel data-adaptive objective function. We also show that special cases of the proposed approach can be used for image denoising and sharpening. Figure 1 depicts a block diagram of our iterative deblurring method. As shown in Fig. 1 and Algorithm 1, the proposed method consists of a number of steps (outer iterations), such that at each step k, an updated objective function is minimized using Conjugate Gradient (CG) inner iterations to obtain the corresponding estimate $\hat{\mathbf{z}}^{(k)}$. To clarify the differences and contributions of this work as compared to some other nonlocal regularization works [8], [9], [23]–[29], it is worthwhile listing them here.

1) We propose a new cost function (16) for image restoration based on a new definition of the *normalized* graph Laplacian. The proposed cost function (16) includes a normalized regularization term derived from this new definition of the graph Laplacian as well as a new data fidelity term. The normalizing coefficients are obtained from a fast symmetry preserving matrix balancing algorithm [30]. This results in some desired spectral properties for the graph Laplacian. Namely, the proposed Laplacian is symmetric, positive semi-definite, and when applied to a constant vector, it returns the zero vector. In this paper, we will discuss the spectral properties of the proposed graph Laplacian and compare its attributes and performance with the existing graph Laplacians.
2) Taking advantage of the quadratic form of the proposed cost function as well as spectral properties of the proposed Laplacian matrix, we present a filtering interpretation for different terms in our objective function as a tool for spectral analysis of the resulting restoration algorithms. Moreover, the symmetric and sparse nature of the resulting filtering and Laplacian matrices equips us with the required tools for efficient implementation of the algorithm using CG and fast sparse matrix-vector products.
3) The way we initially compute the kernel similarity values is different in the sense that we start with a once denoised version of the input noisy and blurry image, and hence avoid the contribution of the noise and ringing artifacts of other deblurring algorithm to the computation of the similarity weights as in see [9]. This denoised initial image is also exploited as a plug-in estimator of $A\mathbf{z}$ (blurred clean image) used in the Predicted Mean Squared Error (PMSE) measure for stopping the inner CG iterations. We allow the weights to be updated during the outer iterations to improve the performance of the algorithm by computing the similarity weights from the enhanced versions of the input image through the outer iterations.
4) We have introduced sharpening and denoising methods as special cases of the proposed general cost function. The denoising aspect is a novelty of the technique that we have previously explored in depth in [31], but it is worth mentioning again here because it fits the proposed general framework.
5) Our approach is quite general in the sense that it is able to handle a variety of different PSFs including symmetric PSFs and more challenging motion blur PSFs.

We emphasize that as compared to [32] specifically, this paper is different in the following respects:

1) Although the general symmetrizing idea is similar, [32] focuses on symmetrizing smoothing filters, in which it starts from a non-symmetric smoothing filter and returns its symmetrized version using the original Sinkhorn matrix balancing algorithm in [33] with performance and analysis advantages described in [32]. In this paper, we start from the symmetric and non-negative similarity matrix K and use a different fast matrix balancing algorithm, designed for scaling symmetric and non-negative matrices, with fast convergence and

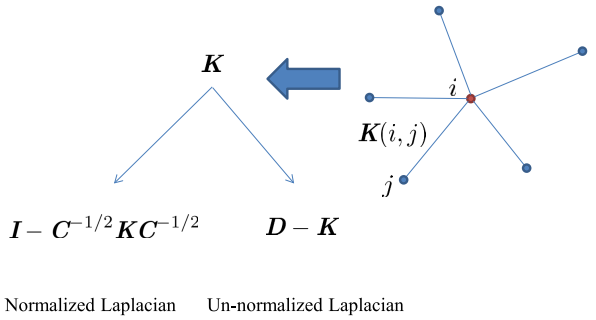


Fig. 2. Graph representation of images and construction of kernel similarity matrix K, un-normalized Laplacian $D-K$ and normalized Laplacian $I-C^{-1/2}KC^{-1/2}$.

symmetry-preserving properties even when the matrix scaling algorithm is stopped early [30].

2) We use the symmetric and doubly stochastic output of [30] to define the normalized Laplacian and we use it in a variational graph-based formulation for various restoration problems.

In the rest of the paper, we elaborate on the above mentioned properties. In Section II, we summarize some of the existing nonlocal regularization restoration methods in a unified graph representation framework. In Section III, we discuss how to derive the symmetric kernel similarity and filtering matrices as the main building blocks of our algorithm. In addition, we present an appropriate definition of normalized Laplacian matrix for filtering purposes and discuss its spectral properties. Section IV is devoted to introducing the objective function and the proposed procedure to optimize it in order to get the final estimate. Section V discusses special cases of the proposed objective function introduced in Section IV for image denoising and sharpening. In Section VII, we verify the effectiveness of the proposed deblurring algorithm via a number of synthetic and real experiments of deblurring color images for both symmetric PSFs (Gaussian and out-of-focus PSFs) as well as nonlinear camera motion blurs. Final conclusion and discussion is provided in Section VIII. Throughout the paper, vectors are represented by boldface small letters, and matrices are shown by capital letters. Also, in iterative update equations, subscript indices for vectors indicate inner iteration numbers, whereas superscript indices represent outer iteration numbers.

II. Related Work

In this section, we summarize some of the existing methods based on the idea of nonlocal regularization in a graph-based framework. We first clarify our notation and summarize some of the definitions mostly common in the nonlocal regularization approaches in the literature. As depicted in Fig. 2, any image can be defined as an intensity function on the vertices V of a weighted graph $G=(V,E,K)$ consisting of a finite set V of vertices (image pixels) and a finite set $E\subset V\times V$ of edges (i,j) with the corresponding weights $K(i,j)$ which measure similarity between vertices (pixels) i and j in the graph (e.g., Eq. 11). The function (intensity) values of the image can be denoted as a vector[1] $\mathbf{z}=[z(1),\ldots,z(N)]^T$. The similarity weights are represented as an $N\times N$ matrix K, which is symmetric and non-negative.

As shown in Fig. 2, graph Laplacian matrix is derived from K and plays an important role in describing the underlying structure of the graph signal. There are three different definitions of the graph Laplacian commonly used in the literature in the context of graph signal and image processing, each having different spectral properties [34]–[36]. In this paper, we present a fourth one, a new normalized graph Laplacian for image processing purposes. In Table I, we have summarized the properties of different types of Laplacians used in the literature along with those of our proposed definition.

In [23] and [37], the difference of a function $z: V\rightarrow\Re$ on an edge $(i,j)\in E$ of the graph G is defined as:

$$(dz)(i,j)=\sqrt{K(i,j)}(z(j)-z(i)). \tag{3}$$

Also, the weighted gradient vector of a function z at a vertex $i\in V$ can be expressed as:

$$\nabla z(i)=[dz(i,j_1),\ldots,dz(i,j_m)]^T,\quad \forall(i,j)\in E. \tag{4}$$

Accordingly, the Laplace operator of z at a vertex i is derived as:

$$\Delta z(i)=\sum_{j,j\sim i}K(i,j)(z(i)-z(j)). \tag{5}$$

where $j\sim i$ stands for the vertices j in the graph such that j is connected to i; i.e., $(i,j)\in E$.

The authors in [23] and [37] propose a nonlocal regularization approach by considering the following cost function:

$$E(\mathbf{z},\mathbf{y},\eta)=\|\mathbf{z}-\mathbf{y}\|^2+\eta R(\mathbf{z}), \tag{6}$$

where the regularization functional R is:

$$R(\mathbf{z})=\frac{1}{2}\sum_{i=1}^{N}\|\nabla z(i)\|^2=\frac{1}{2}\sum_{i=1}^{N}\sum_{j,j\sim i}K(i,j)(z(i)-z(j))^2. \tag{7}$$

It essentially enforces the similar pixels of the image -as measured by the function $K(.,.)$- to remain similar in the final estimate. By minimizing the above cost function with respect to the unknown $\mathbf{z}$, they recover the desired image. Note that the regularization term (7) can be expressed based on the un-normalized graph Laplacian $D-K$ as [23], [34], and [38]:

$$R(\mathbf{z})=\mathbf{z}^T(D-K)\mathbf{z}, \tag{8}$$

where $D=diag\{K\mathbf{1}_N\}$ is a diagonal matrix whose ith diagonal element is the sum of the elements of the ith row of K, and $\mathbf{1}_N$ is the N-dimensional vector of all ones. In [23] and [37], the authors also introduce the Laplace operator associated to the traditional normalized graph Laplacian $I-D^{-1/2}KD^{-1/2}$. However, they do not use this definition of the Laplacian in their formulation because of the fact that the output of this operator is not null when the input is constant.

In [25], Gilboa and Osher consider the same nonlocal functional as the one in (6) for image denoising. They discuss the case $\eta=\infty$ as well, for which they derive the diffusion flows (iterations) defined based on the un-normalized

[1] Note that $N=n^2$.

TABLE I
PROPERTIES OF DIFFERENT GRAPH LAPLACIANS. LAST ROW IS OUR DEFINITION

Reference	Graph Laplacian	Symmetric	DC eigenvector	Stochastic property
[23], [25]	$D-K$	Yes	Yes	No
[27]	$I-D^{-1/2}KD^{-1/2}$	Yes	No	No
[28], [29]	$I-D^{-1}K$	No	Yes	$D^{-1}K$ is row-stochastic
ours	$\mathbf{I-C^{-1/2}KC^{-1/2}}$	**Yes**	**Yes**	**W is doubly-stochastic**

Laplacian $D-K$. They show these types of iterations give better performance than the diffusion iterations corresponding to another type of normalized Laplacian (called random walk Laplacian and defined as $I-D^{-1}K$). Also, in [26], the authors introduce the gradient-based and difference-based regularizing functionals, respectively as (we consider here their discrete versions):

$$J(\mathbf{z}) = \sum_{i=1}^{N} \phi(\|\nabla z(i)\|^2) = \sum_{i=1}^{N} \phi\Big(\sum_{j,j\sim i} K(i,j)(z(j)-z(i))^2\Big), \tag{9}$$

$$J_a(\mathbf{z}) = \sum_{i=1}^{N} \sum_{j,j\sim i} \phi(K(i,j)(z(j)-z(i))^2), \tag{10}$$

in which $\phi(s)$ is a positive function, convex in $\sqrt{s}$, with $\phi(0)=0$. They consider the quadratic case $\phi(s)=s$, where the above functionals coincide. They also investigate the case $\phi(s)=\sqrt{s}$, for which nonlocal TV and anisotropic nonlocal TV functionals are derived from the gradient-based and difference-based approaches, respectively. They have applied their framework to inpainting and detecting and removing irregularities from textures.

In [28], Szlam, Maggioni, and Coifman propose function adapted diffusion processes (using the random walk Laplacian $I-D^{-1}K$). They also propose a filtering procedure using a type of thresholding of the expansion coefficients of the input function on the linearly independent bases of the operator $D^{-1}K$. Reference [29] is also based on the same idea (expansion of the input data on the space spanned by the right eigenvectors of random walk Laplacian) for surface smoothing with weights derived *locally* in a non data-adaptive manner. The denoising method in [27], exploits a similar filtering idea as in [28]; i.e., constructing a weighted graph from the input image characterized by its normalized Laplacian $I-D^{-1/2}KD^{-1/2}$, and expanding the noisy image using the orthonormal bases of the normalized graph Laplacian, then hard thresholding of the transform coefficients to derive the corresponding estimates for image pixel intensities in different patches of the image. In [24], a patch-based functional is considered for denoising 3D image sequences acquired via fluorescence microscopy. This functional is based on minimizing a difference penalty term which is defined using the weighted difference between its patches rather than the weighted difference between its pixels. The minimizer of such a cost function can be equivalently expressed as a nonlocal filtering process; i.e., $\hat{\mathbf{z}}=D^{-1}K\mathbf{y}$.

Finally, the most relevant paper to our work is reference [9], in which Zhang *et al.* propose two efficient algorithms for solving nonlocal TV-based image deconvolution.[2] They also provide a weight updating strategy within these iterative methods which was found to be *ineffective* in improving the performance of their algorithms. Therefore, they chose to compute the similarity weights only once from the simple Tikhonov regularization based deblurring estimate. Also, [8] proposes a regularization technique using total variation on nonlocal graphs for inverse problems, when the input data has undergone linear degradation as well as additive noise. Note that our deblurring algorithm uses a different nonlocal approach, in which the corresponding regularization term is defined using the normalizing coefficients derived from Sinkhorn's algorithm in [30]. Moreover, based on our experiments, the weight updating strategy is indeed effective in improving the performance of the proposed algorithm within the same quadratic framework. Furthermore, according to the analysis provided in [22], using the data fidelity term involving different derivatives of the residual is better able to model the underlying process for deblurring problems (especially for real motion blurred images).

III. DERIVATION OF BUILDING BLOCK MATRICES OF THE PROPOSED ALGORITHM

In this section, we introduce the kernel similarity matrix K and a closely related doubly stochastic symmetric matrix W as the main filtering building block of our iterative algorithm from a graph point of view. Having these matrices at hand, we can define the normalized Laplacian matrix whose spectral properties are crucial for analyzing the behavior of the algorithm.

A. Kernel Similarity Matrix K and Filtering Matrix W

While our approach is general enough to include any valid kernel similarity function [39], [40], the (i,j)th element of the kernel similarity matrix K is computed here using the nonlocal means (NLM) definition as [41]

$$K(i,j) = \exp\Big(-\frac{\|\hat{\mathbf{z}}_i - \hat{\mathbf{z}}_j\|^2}{h^2}\Big), \tag{11}$$

in which $\hat{\mathbf{z}}_i$ and $\hat{\mathbf{z}}_j$ are patches around the pixels i and j of the image $\hat{\mathbf{z}}$, and h is a smoothing parameter. Note that at each outer iteration, the kernel similarity weights are re-computed from the estimate at the previous iteration. As a result of the

[2] As mentioned, the corresponding regularization term is derived using $\phi(s)=\sqrt{s}$ in Eq. (9).

above definition for the kernel similarity weights, the matrix K would be a symmetric non-negative matrix. Furthermore, we only compute the similarity between each patch and a small neighbourhood of patches around it (e.g., a search window of size 11×11 of patches around each patch). Therefore, the matrix K is sparse. This sparse structure is appealing from a computational point of view.

Applying Sinkhorn matrix balancing procedure [33] to the matrix K yields the doubly stochastic filtering matrix W. We use a recent fast version of the original algorithm for symmetric non-negative matrices [30]. This balancing algorithm returns a diagonal scaling matrix C, such that the resulting matrix $W = C^{-1/2}KC^{-1/2}$ is a symmetric non-negative doubly stochastic matrix. Since W is symmetric, it can be decomposed as $W = VSV^T$, where V is an orthonormal matrix whose columns are the eigenvectors of W, and $S = diag\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ is a diagonal matrix consisting of eigenvalues of W as its diagonal elements. Moreover, since W is doubly stochastic, it has eigenvalues in the range $\lambda_1 = 1 > \lambda_2 \geq \cdots \geq \lambda_N \geq 0$ [40]. The largest eigenvalue is exactly equal to 1 with the corresponding DC eigenvector $\mathbf{v}_1 = (1/\sqrt{N})[1, 1, \ldots, 1]^T = (1/\sqrt{N})\mathbf{1}_N$ [40]. Intuitively, it means that applying W to a signal, preserves the DC component of the signal. This is a desirable property for filtering purposes. Note that the spectral analysis of the matrix W reveals its inherent low-pass nature (the largest eigenvalue corresponds to the DC component) [32].

B. Normalized Graph Laplacian Matrix

At this point, we define our normalized graph Laplacian matrix as

$$I - W = I - C^{-1/2}KC^{-1/2}. \tag{12}$$

This is the proper definition of the normalized graph Laplacian matrix for image filtering purposes, as opposed to its traditional definition in graph theory literature $I - D^{-1/2}KD^{-1/2}$ [34], [38]. It is worthwhile comparing this traditional definition of the normalized Laplacian with our proposed definition which is based on a very different scaling of the similarity matrix K using matrix balancing [30]. Our definition of the normalized Laplacian ($I - W = I - C^{-1/2}KC^{-1/2}$) is symmetric, positive semi-definite, with the zero eigenvalue associated to the constant eigenvector $\frac{1}{\sqrt{N}}\mathbf{1}_N$. Hence, when applied to a constant function, it returns a zero vector. The traditional definition of the normalized graph Laplacian lacks the desired filtering property of having DC eigenvector as one of the basis eigen functions [34]. As a result, the definition of normalized graph Laplacian in (12) is proposed and used in this paper. This definition has the desired spectral properties for our specific applications as well as a nice filtering interpretation. In fact, the set of eigenvectors of $I - W$ can be considered as the basis functions of the underlying graph, and its eigenvalues can be thought of as the corresponding graph frequencies. Also, note that the Laplacian $I - W$ has a high-pass filtering nature (with null eigenvalue corresponding to the DC eigenvector). This property is consistent with the expected behavior of the Laplacian filter in image processing. Consequently, when applied to an image, $I - W$ can be directly interpreted as a data-adaptive Laplacian filter. Therefore, it enables us to incorporate different types of filters in the data term coupled to the regularization term based on the application at hand. In the "random walk" Laplacian $I - D^{-1}K$, (from the theory of Markov chains), the (i, j)th element of $D^{-1}K$ represents the probability of moving from node i to node j of the graph in one step, given that we are in node i [42]. A similar random walk interpretation can be provided by our symmetric doubly stochastic filtering matrix $W = C^{-1/2}KC^{-1/2}$, with analysis and performance advantages over $D^{-1}K$ for image filtering, as discussed in [32].[3] Furthermore, for image deblurring applications, another advantage is that our resulting linear equations are symmetric and positive definite, providing us with fast methods for solving large linear systems of equations with optimization methods like CG. Compared to the un-normalized graph Laplacian, as we have shown in [31], our approach based on the proposed normalized graph Laplacian $I - W$ provides better performance.

In order to better demonstrate the different expressions of the difference and Laplacian operators as well as the regularization term corresponding to our normalized Laplacian, we derive them here. We can define the difference operator corresponding to the proposed normalized graph Laplacian as:

$$dz(i, j) = \sqrt{K(i, j)}\left(\frac{z(j)}{\sqrt{C(j, j)}} - \frac{z(i)}{\sqrt{C(i, i)}}\right), \tag{13}$$

where $C(j, j)$ and $C(i, i)$ are the corresponding j and ith diagonal elements of the diagonal matrix C derived from the matrix balancing algorithm [30], [32]. From the above equation along with the definition of the divergence operator [23], the Laplace operator corresponding to the normalized Laplacian $I - C^{-1/2}KC^{-1/2}$ is:

$$\triangle z(i) = \frac{1}{\sqrt{C(i, i)}} \sum_{j, j \sim i} K(i, j)\left(\frac{z(i)}{\sqrt{C(i, i)}} - \frac{z(j)}{\sqrt{C(j, j)}}\right), \tag{14}$$

As a result, our proposed regularization term can be written as:

$$\begin{aligned} R(\mathbf{z}) &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j, j \sim i} K(i, j)\left(\frac{z(i)}{\sqrt{C(i, i)}} - \frac{z(j)}{\sqrt{C(j, j)}}\right)^2 \\ &= \mathbf{z}^T(I - W)\mathbf{z}. \end{aligned} \tag{15}$$

Note that the Laplace operator in (14) describes the effect of our normalized Laplacian at each pixel i, when applied to an input vector $\mathbf{z}$. As the Laplace operator is a second order derivative operator, the name Laplacian for the corresponding matrix operator is appropriate, and common in graph theory. In the next section, we will describe how to use the proposed graph Laplacian to develop a new restoration algorithm.

IV. Proposed Deblurring Method

As depicted in Fig. 1, the proposed algorithm consists of inner and outer iterations. The reason is that for computing the data-adaptive matrix K, a good rough estimate of the

[3] In fact, W can be thought of as the transition probability matrix of the Markov chain defined on the graph.

underlying unknown image is needed. This estimate is gradually improved as we proceed through iterations. In each outer iteration, the matrix W is computed once and used to define the following objective function to be minimized with respect to the unknown image $\mathbf{z}$

$$E(\mathbf{z}) = (\mathbf{y} - A\mathbf{z})^T\{I + \beta(I - W)\}(\mathbf{y} - A\mathbf{z}) + \eta\ \mathbf{z}^T(I - W)\mathbf{z}, \tag{16}$$

where $\beta \geq -1$ and $\eta > 0$ are the parameters to be tuned based on the amount of noise and blur. Note that in the above objective function, data and prior terms are coupled via the matrix W. This coupling is controlled by means of the parameter β. The first term favors a solution $\mathbf{z}$ such that its blurred and then filtered version is as close as possible to the filtered version of the input $\mathbf{y}$. Frequency selectivity of this common filter is determined by the parameter β according to the amount of the noise and blur. The second term is essentially a data-adaptive difference term favoring certain smoother solutions based on the structure of the underlying data encoded in the normalized Laplacian matrix $I - W$, defined in the previous section.

Let us take a look at the cost function in (16) from a filtering point of view. This filtering interpretation provides a more intuitive perspective on the objective function. For this purpose, Eq. (16) is rewritten in the following form

$$E(\mathbf{z}) = \|\{I + \beta(I - W)\}^{1/2}(\mathbf{y} - A\mathbf{z})\|^2 + \eta\|(I - W)^{1/2}\mathbf{z}\|^2. \tag{17}$$

Note that $I + \beta(I - W) = V\Lambda V^T$ is a symmetric and positive semi-definite matrix. Therefore, the matrix $\{I + \beta(I-W)\}^{1/2} = V\Lambda^{1/2}V^T$ has a filtering behavior similar to that of $I+\beta(I-W)$. Once we have the eigendecomposition of the filtering matrix W, the ith diagonal element of the matrix Λ can be written in terms of the associated ith diagonal element of S (that is λ_i) as $1 + \beta(1 - \lambda_i)$. Since the matrix $I - W$ is a high-pass filter, with $\beta > 0$, $I + \beta(I - W)$ behaves like a sharpening filter on the residuals $\mathbf{y} - A\mathbf{z}$, and so does $\{I + \beta(I - W)\}^{1/2}$. According to the analysis provided in [22], using the data fidelity term involving different derivatives of the residual is better able to model the underlying phenomenon for deblurring problems (especially for real images).

The same analysis applies to the second term in (17), where both Laplacian $I-W$, and its square root $(I-W)^{1/2}$, are adaptive high-pass filters. Consequently, the resulting regularization expression in (17) adaptively penalizes high frequencies in the final solution to avoid unpleasant artifacts due to the noise amplifications and ringing artifacts while maintaining fine details in the restored image.

In order to minimize the cost function in (16) at each step, the corresponding gradient is set equal to zero as

$$\nabla E(\mathbf{z}) = -2A^T\{I + \beta(I - W)\}(\mathbf{y} - A\mathbf{z}) + 2\eta(I - W)\mathbf{z} = \mathbf{0}, \tag{18}$$

which results in the following symmetric positive definite system of liner equations

$$(A^T\{I + \beta(I-W)\}A + \eta(I - W))\mathbf{z} = A^T\{I+\beta(I-W)\}\mathbf{y}. \tag{19}$$

TABLE II
CONDITION NUMBER OF $(A^T(I + \beta(I - W))A + \eta(I - W))$ FOR DIFFERENT VALUES OF η AND β AND BLURRING MATRIX A CORRESPONDING TO OUT-OF-FOCUS BLUR WITH RADIUS 7. THE CONDITION NUMBER OF A^TA IS 5.74×10^{20}

	$\eta = 0.05$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
$\beta = 0.1$	2.02×10^3	1.17×10^3	705	459
$\beta = 0.5$	1.97×10^3	1.15×10^3	700	402
$\beta = 0.7$	1.95×10^3	1.14×10^3	699	543

Conjugate Gradient is then used to solve the above system. Also, note that A and A^T are interpreted as blurring with the PSF or its flipped version, respectively. Our experiments show that three outer iterations suffice to get the desired deblurred output in most cases. Also, note that the only restriction on the parameter β is that it should be selected such that the corresponding system of linear equations in (19) remains positive definite. The matrix $I + \beta(I - W)$ is also required to be positive semi-definite for the existence of its square root in the data fit term in (17). A sufficient condition is $\beta \geq -1$.

A. Spectral Analysis of the Overall Deblurring Algorithm

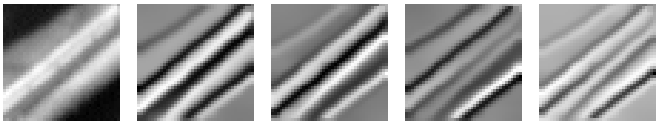
For analysis purposes, we are able to provide a filtering interpretation of the final estimate at each outer step of the algorithm. Note that the minimizer of the cost function in (17) can be expressed as:

$$\hat{\mathbf{z}} = F(A, W)A^T(I + \beta(I - W))\mathbf{y}, \tag{20}$$

where

$$F(A, W) = \{A^T(I + \beta(I - W))A + \eta(I - W)\}^{-1}. \tag{21}$$

Eq. (20) can be interpreted as (1) filtering $\mathbf{y}$ by $I+\beta(I-W)$, (2) back projection through multiplication by the transpose of the blurring matrix A, and (3) applying the symmetric matrix $F(A, W)$. In other words, if we consider the spectral decomposition of this symmetric matrix as $F(A, W) = \Theta\Upsilon\Theta^T$, the columns of the matrix Θ serve as an orthonormal basis for filtering the vector $A^T(I + \beta(I - W))\mathbf{y}$, thereby providing a spectral filtering interpretation for the corresponding deblurring solution at each outer step of the algorithm. Since an inverse operation is involved in (20), we consider a simple experiment investigating the condition number of the matrix $A^T(I + \beta(I - W))A + \eta(I - W)$. For this purpose, we use the MATLAB code in [44] to explicitly construct the blurring matrix A related to out-of-focus blur with radius 7. Table II illustrates the condition number of $A^T(I + \beta(I - W))A + \eta(I - W)$ for different values of the parameters η and β. The condition numbers of $A^T(I + \beta(I - W))A + \eta(I - W)$ for different values of η in comparison to the condition number of A^TA show the effectiveness of our procedure for regularizing the ill-posed deblurring problem and the corresponding linear equations. Also, the basis eigenvectors in Θ corresponding to the four largest eigenvalues of $F(A, W)$ are depicted in Fig. 3. As can be seen in Fig. 3, the eigenvectors associated with the largest eigenvalues of $F(A, W)$ indicate the data-adaptive nature of the corresponding filter.



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

Fig. 3. (a) Original 41×41 image, and (b), (c), (d), (e) the eigenvectors of $F(A, W)$ corresponding to the four largest eigenvalues for $\beta = 0.7$ and $\eta = 0.2$.

V. Special Cases of the Proposed Objective Function

It is interesting to consider two special cases of the above objective function in (16) for two different applications, namely denoising and sharpening.

A. Image Denoising

When $A = I$ in (16), the problem reduces to that of image denoising. This case has been discussed in our previous work [31]. It turns out that $\beta = -1$, is the appropriate choice for image denoising [31]. Also, the optimal value of the regularization parameter η is selected using a SURE-based estimated MSE approach [45]. The proposed denoising formulation is able to describe some of the existing kernel similarity-based denoising algorithms, and provides an iterative approach for their further improvement [31].

B. Image Sharpening

Another special case is when image contains a moderate blur, but no information about the blurring process is available. In such cases, one can resort to the following cost function

$$E(\mathbf{z}) = (\mathbf{y} - \mathbf{z})^T \{I + \beta(I - W)\}(\mathbf{y} - \mathbf{z}), \tag{22}$$

which comes from Eq. (16), by setting $A = I$ and $\eta = 0$.

Optimizing the above objective function using simple steepest descent, yields:

$$\hat{\mathbf{z}}_\ell = \hat{\mathbf{z}}_{\ell-1} + \mu\{I + \beta(I - W)\}(\mathbf{y} - \hat{\mathbf{z}}_{\ell-1}). \tag{23}$$

By selecting the step size parameter $\mu = 1$, and with zero initialization of the SD iterations in (23); i.e., $\hat{\mathbf{z}}_0 = \mathbf{0}$, the first iteration takes the form

$$\hat{\mathbf{z}}_1 = \{I + \beta(I - W)\}\mathbf{y}. \tag{24}$$

For $\beta > 0$, Eq. (24) can be interpreted as data-adaptively adding to the input image some amount of its high-pass filtered version. This procedure results in a sharper image. Although there is no access to the exact PSF, since the matrix W is computed from the input blurred image, it contains some information about the original image as well as the blurring process. Therefore, Eq. (24) provides us with a data-adaptive sharpening (or to say rough deblurring) technique.

VI. Implementation Details

The first step of the iterative algorithm is to compute the kernel similarity matrix[4] K. At each outer iteration k, we compute this matrix from the final estimate of the previous step, i.e., from $\hat{\mathbf{z}}^{(k-1)}$, as shown in Fig. 1. The values of the regularization parameters η and β are selected based on the noise variance and blurring scenario, and are kept fixed at each step of the algorithm, for all the test images. For deblurring examples, for instance, the parameter β lies in the range $(0, 1)$, and the parameter η is empirically selected in the range $(0, 0.4)$. The closer is β to 1, the larger is the effect of the data-adaptive high-pass filter $I + \beta(I - W)$ in the data term of the cost function in (16), which results in encouraging higher frequencies of $A\mathbf{z}$ to be close to those of $\mathbf{y}$. Similarly, the larger is the value of η, the more penalty is put on the norm of the high-pass filtered version of the desired solution $\mathbf{z}$. For instance, larger values of η and smaller values of β are used when the amount of noise is high in the input image, and the image is moderately blurred. Similarly, when the amount of noise is low while the image is severely blurred, larger values of β and smaller values of η are used. These tunings are done for each scenario of blur and noise for a set of test images to have visually pleasant results, and are kept fixed for all other input images with the same degree of degradation, as shown in the next section.

To further speed up the convergence of the iterative algorithm, each step of the algorithm is initialized with the corresponding estimate from the previous step. In experiments, in order to avoid noise amplification and ringing artifacts, the maximum number of inner and outer iterations are set beforehand based on the amount of degradation,[5] and then the iterations are stopped using a rough estimate of Predicted-MSE (PMSE) measure as[6]

$$\widehat{PMSE}(q, k) = \frac{1}{n^2}\|\widehat{A\mathbf{z}} - A\hat{\mathbf{z}}_k^{(q)}\|^2, \tag{25}$$

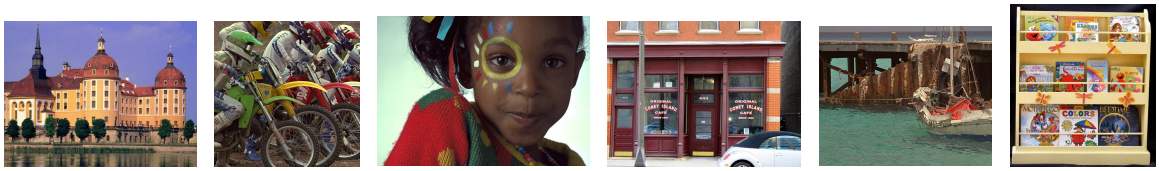
where $\widehat{A\mathbf{z}}$ is an estimate of the blurred clean image $(A\mathbf{z})$, which is derived by denoising input noisy blurry image, and $\hat{\mathbf{z}}_k^{(q)}$ is the corresponding estimate of the desired image at the kth CG iteration of the qth outer iteration. That is, we stop CG iterations whenever $\widehat{PMSE}(q, k+1) > \widehat{PMSE}(q, k)$.

There are two main computational burdens for the algorithm. First is the computation of the kernel similarity coefficients, where its special form allows us to take advantage of the idea of integral images [47]. This technique is very effective to reduce the computational complexity of the algorithm. Second is the matrix-vector products required at each iteration of CG method for optimizing the objective function in (16). However, because of the special structure of the matrices involved, it is possible to implement the algorithm using Fast Fourier Transform (FFT) and fast

[4]This is computed from the denoised version of the input image at the beginning of the algorithm.

[5]For more blurry images, we need more iterations for convergence of CG iterations. Also, as we initialize the CG iterations with more enhanced images as we proceed through the outer iterations, the number of inner iterations is decreased by a step nDec, as the number of outer iterations increases.

[6]Predicted-MSE is defined as $PMSE(q, k) = \frac{1}{n^2}\|A(\mathbf{z} - \hat{\mathbf{z}}_k^{(q)})\|^2$ [46].



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

Fig. 4. Set of color images used for evaluation of our method: (a) Building image (480 × 640), (b) Motocross bikes image (494 × 494), (c) Girl image (496 × 700), (d) Street image (480 × 640), (e) Boat image (420 × 520), and (f) Book shelf image (580 × 520).

sparse matrix-vector products. It is also possible to exploit the symmetric structure of the kernel similarity matrix K (and of course that of W) to reduce memory requirements.

VII. Numerical Experiments

In this section, the effectiveness of our iterative approach is verified through a number of synthetic and real experiments. Throughout the deblurring experiments, our focus is on more practical cases with severe blur and small amount of noise in the captured images. We have set up experiments for Gaussian, out-of-focus, and nonlinear camera motion blur. For all cases, we have compared the performance of our algorithm with some of the best existing non-blind deblurring algorithms. Also, for both motion and out-of-focus blurs, the iterative algorithm is applied to real images to evaluate its performance for such more complicated cases. Since the proposed method is a non-blind deblurring algorithm, for real deblurring examples, we use PSFs derived from other existing blur kernel estimation methods. For this purpose, in case of real out-of-focus blur, the PSF is estimated using "deconvblind" MATLAB function. In case of real motion deblurring, the estimated PSFs from [22] and [48] are used. Furthermore, synthetic and real image sharpening examples are provided. For color images, the proposed deblurring algorithm is applied independently to R, G, and B channels of the input color image to get the final estimate. In all the experiments, object oriented MATLAB functions in [49] are used for performing matrix-vector products of the form $A\mathbf{z}$ and $A^T\mathbf{z}$. PSNR in dB and the SSIM index are used for comparison purposes [50]. SSIM index is shown to be a more reliable metric for comparison of deblurring algorithms than the widely used PSNR measure [50]. In order to show the effectiveness of our proposed method compared to one of the existing approaches based on nonlocal means regularization for image deconvolution [9], we first demonstrate deblurring examples related to this comparison.

A. Comparison With Nonlocal Means Regularization Deconvolution Algorithm

In the following experiments, Cameraman image is circularly convolved with box average and motion blurs, and white Gaussian noise with standard deviation $\sigma = 1$ is added to generate the noisy blurred examples. The regularization parameters of both algorithms are selected for the best performance in each case.[7] For the next deblurring example with nonlinear motion blur and the same noise level ($\sigma = 1$), the regularization parameter μ of the algorithm in [9] and the parameters η and β in our algorithm are changed and the other parameters in both algorithms remain the same as the first experiment. Note that for fair comparisons, we are using the same definition of the nonlocal similarity function as the one in [9] and [41] with the corresponding Gaussian weighting for the patch elements.

For box average blur kernel, as can be seen from the results in Fig. 5, while our method gives slightly better visual quality, both algorithms show comparable quantitative performance. However, for nonlinear motion blur, the algorithm in [9] almost fails with the current settings provided by the authors as shown in Fig. 6, while our proposed algorithm shows superior performance in this case in terms of both PSNR and SSIM values. This is due to our definition of the cost function, and the new normalized regularization term as well as a different data fit term. This good performance can also be attributed to the way we initialize the algorithm, such that we do not compute the weights from the output of another deblurring algorithm to avoid contributing the deblurring artifacts to the similarity weights. Instead, we allow the weights to be updated as we proceed through iterations.

B. Symmetric Blurs

Two kinds of symmetric blur are considered for these examples: Gaussian blur and out-of-focus blur. A 25 × 25 Gaussian blur with standard deviation 1.6 is convolved with a set of color images[8] shown in Fig. 4. Also, out-of-focus blur is produced using a disk function with radius 7 and is used to generate the corresponding blurred examples. Then, additive white Gaussian noise with variances equal to 0.2 and 1 is added to the blurred images. We compare the performance of our algorithm with that of IDDBM3D algorithm in [18]. Periodic boundary conditions are used in these examples [1]. Also, we use patch size of 5 × 5, search neighborhood size of 11 × 11, number of outer iterations equal to 3, and the step decrement of the number of inner iterations (nDec) equal to 30 in these experiments. The values of the parameters η, β, h, and maximum number of inner CG iterations have been summarized in Table IV. Note that, the parameters of both algorithms are set for best performance in each case for this set of images, and for the given blurs and noise variances.

Figures 7 and 8 depict deblurring outputs of our algorithm compared to those of IDDBM3D for noise variance of 0.2

[7]We use the MATLAB and mex codes provided by the authors in http://www.math.ucla.edu/~xqzhang/html/code.html.

[8]Test images are from Kodak Lossless True Color Image Suite (http://r0k.us/graphics/kodak/) and the web page for [51].

(a) (b) (c) (d)

Fig. 5. Deblurring example with noisy blurred Cameraman image with 9×9 box average blur kernel and additive white Gaussian noise with standard deviation $\sigma = 1$: (a) clean image, (b) blurred noisy image, (c) output of [9] with regularization parameter $\mu = 14$ in the algorithm (PSNR = 27.43dB, SSIM = 0.8544), and (d) output of our proposed deblurring algorithm with $\eta = 0.02$ and $\beta = 5 \times 10^{-4}$ (PSNR = 28.02dB, SSIM = 0.8537).

(a) (b) (c) (d)

Fig. 6. Deblurring examples with blurred noisy Cameraman image with nonlinear motion blur and additive white Gaussian noise with standard deviation $\sigma = 1$: (a) clean image, (b) blurred noisy image, (c) output of [9] with the regularization parameter $\mu = 80$ in the algorithm (PSNR = 16.82dB, SSIM = 0.3969), and (d) output of our proposed deblurring algorithm with $\eta = 0.022$ and $\beta = 0.2$ (PSNR = 27.94dB, SSIM = 0.8581).

TABLE III

SSIM AND PSNR PERFORMANCE OF THE KERNEL SIMILARITY-BASED ALGORITHM AND IDDBM3D [18] FOR GAUSSIAN BLUR KERNEL OF SIZE 25×25 WITH STANDARD DEVIATION 1.6 AND OUT-OF-FOCUS BLUR GENERATED USING DISK FUNCTION OF RADIUS 7. IN EACH CELL, THE FIRST NUMBER DENOTES SSIM VALUE, AND THE SECOND NUMBER REPRESENTS PSNR VALUE IN DB

Blur	Building		Motocross Bikes		Girl		Street		Boat		Book shelf	
	ours	[18]	ours	[18]	ours	[18]	ours	[18]	ours	[18]	ours	[18]
Gaussian	0.9726	0.9793	0.9760	0.9791	0.9693	0.9756	0.9732	0.9778	0.9642	0.9655	0.9827	0.9841
($\sigma^2 = 0.2$)	28.57	29.30	27.56	28.36	33.18	33.84	28.80	30.29	29.11	29.49	27.25	28.15
out-of-focus	0.9511	0.9637	0.9542	0.9609	0.9378	0.9489	0.9524	0.9614	0.9282	0.9310	0.9711	0.9709
($\sigma^2 = 0.2$)	28.44	29.64	26.92	28.08	32.28	33.00	29.71	31.35	28.79	28.75	27.60	28.19
Gaussian	0.9576	0.9683	0.9623	0.9675	0.9522	0.9630	0.9624	0.9677	0.9448	0.9504	0.9717	0.9787
($\sigma^2 = 1$)	27.59	28.62	26.24	27.22	32.34	32.97	28.34	29.33	28.00	28.62	26.27	27.31
out-of-focus	0.9178	0.9369	0.8974	0.9177	0.9058	0.9201	0.9050	0.9344	0.8656	0.8839	0.9405	0.9537
($\sigma^2 = 1$)	26.90	27.80	25.04	25.75	30.85	31.42	27.19	29.00	26.77	27.35	25.63	26.64

TABLE IV

SET OF PARAMETERS IN DIFFERENT SYNTHETIC COLOR IMAGE DEBLURRING EXAMPLES IN THIS PAPER. NINNER IS THE MAXIMUM NUMBER OF THE INNER CG ITERATIONS

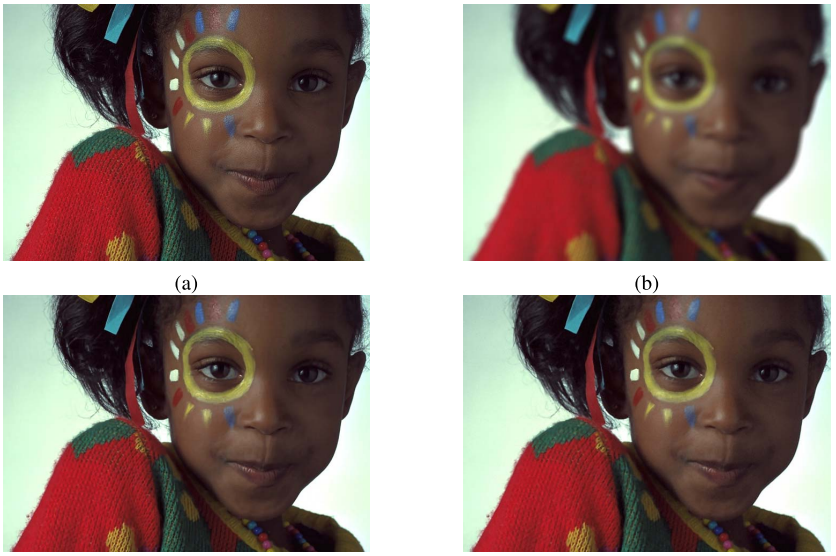
Experiment	η	β	h	nInner
Symmetric ($\sigma^2 = 0.2$)	0.003	0.2	5.5	100
Symmetric ($\sigma^2 = 1$)	0.008	0.001	7.5	100
motion blur ($\sigma^2 = 0.2$)	0.006	0.4	6	100
motion blur ($\sigma^2 = 1$)	0.01	0.01	6.5	80

and synthetic Gaussian and out-of-focus blurs, respectively. Also, Table III summarizes the numerical deblurring results. As can be seen in Table III, our kernel similarity-based algorithm shows very close performance to the state-of-the-art IDDBM3D algorithm in [18] in the case of Gaussian blur. Also, our iterative algorithm performs acceptably in the case of out-of-focus blur. In some cases, our algorithm exhibits slightly better visual quality as can be seen, e.g., in smooth parts of the face of the Girl image in Fig. 8. There is one key difference between our proposed algorithm and IDDBM3D. IDDBM3D is a two step algorithm, in which denoising and deblurring are decoupled. Each step of IDDBM3D essentially involves solving two different objective functions, one for deblurring and the other for denoising. Regarding the computational complexity, even though our algorithm has been written entirely in MATLAB (except the initial denoising step which we use the code provided by the authors in [16]),[9] our algorithm runs faster, making it more appropriate for

[9]In contrast, the computational demanding parts of IDDBM3D have been implemented in C++ using MATLAB mex files.

(a) (b) (c) (d)

Fig. 7. Deblurring example with Gaussian blur: (a) clean Motocross bikes image, (b) blurred noisy image, (c) output of [18], and (d) output of our algorithm.



(a) (b) (c) (d)

Fig. 8. Deblurring example with out-of-focus blur: (a) clean Girl image, (b) blurred noisy image, (c) output of [18], and (d) output of our algorithm.

practical image deblurring applications. To be more specific, for a 480×640 color image, the MATLAB implementation of our kernel similarity-based method runs 4 times faster than the code for IDDBM3D run on a 2.8 GHz Intel Core i7 processor. Furthermore, our method just relies on an initial denoising, whereas IDDBM3D depends on an appropriate estimate from another deblurring algorithm in its grouping phase. In addition, as we demonstrate in the remaining experiments, the proposed method has the flexibility to be applied to a wide variety of blurs including both symmetric and non-symmetric blurs, while IDDBM3D has been designed and tested specifically for symmetric blurs.

Figure 9 shows the output of our algorithm when applied to real noisy and out-of-focus blurred images compared to the outputs of the Focus Magic deblurring software. As can be seen, our algorithm is better able to handle noise amplification related issues. In the following subsections, we consider the effect of different factors on the performance of the proposed deblurring method.

1) Effect of the Patch Size on the Performance of the Proposed Algorithm: In this subsection, we add an experiment investigating the relationship between the patch size and blur kernel width in the deblurring algorithm for the case of out-of-focus blur. Synthetic examples are produced by applying out-of-focus blur kernels with radii 5, 7, 9, and 11 to the Girl image. White Gaussian noise with variance $\sigma^2 = 0.4$ is also added to the blurred images. The corresponding SSIM values are plotted in Fig. 10 versus the patch size for different radii of the out-of-focus blur.[10] It can be seen that for out-of-focus blur, the best performance is not strongly dependent on the patch size regardless of the out-of-focus blur kernel radius. It shows that for such blur kernels, the structure around each pixel is described well just by considering a small 5×5 patch around it. In other words, there is no specific relation between the patch size and the size of the blur kernel. In fact, one can fix this parameter and change other parameters like the regularization parameter η to control the quality of the output image.

2) Effect of the Smoothing Parameter h on the Performance of the Proposed Algorithm: In order to investigate the effect of the parameter h, we apply out-of-focus blur with radius 7 to the clean Girl image and then add noise with two different variances (0.2 and 1) to it. The results are shown in Fig. 11. Note that the same set of parameters are used for both cases

[10] We consider patch sizes 3, 5, and 7.

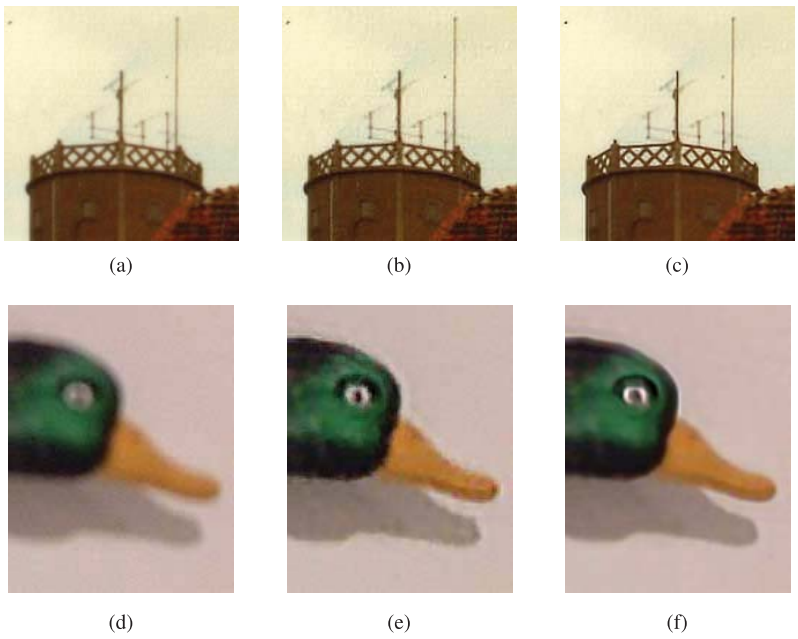


Fig. 9. Real sharpening examples: left column: input blurred noisy image, middle column: output of Focus Magic software at http://www.focusmagic.com, and right column: output of our algorithm ($\eta = 0.1$ and $\beta = 0.06$ for both images).

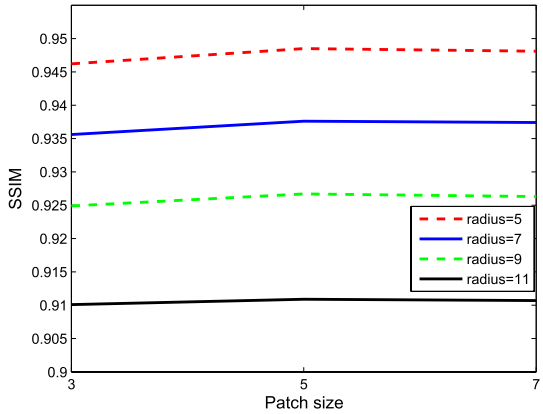


Fig. 10. The relationship between the patch size and blur kernel width for Girl image and out-of-focus blur with different radii.

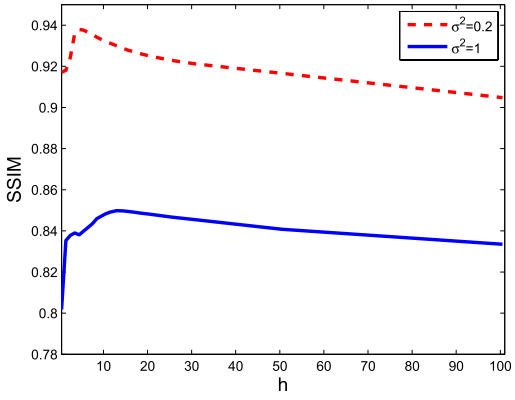


Fig. 11. The effect of the smoothing parameter h in the kernel similarity function for Girl image and out-of-focus blur with radius 7 for two different noise variances.

of noise variance. As can be seen, with the same set of parameters, the optimal smoothing parameter h is greater for the higher noise level. Also, it is evident from Fig. 11 that

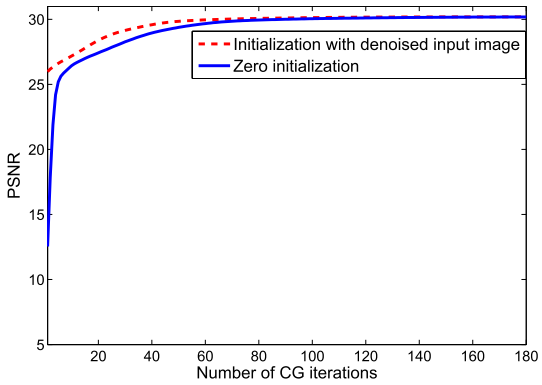


Fig. 12. Convergence plots of the CG iterations for different initializations.

the algorithm is not very sensitive to the selection of this parameter.

3) Effect of the Initialization of the CG Iterations on the Performance of the Algorithm: In order to investigate the effect of the specific initialization for CG iterations at first step of the deblurring algorithm, we consider a simple experiment with Girl image for Gaussian blur with standard deviation 1.6 and noise variance $\sigma^2 = 1$. We consider two different initializations, of CG iterations with (1) the denoised version of the input noisy and blurred image versus (2) initializing it with zero image, i.e., $\hat{\mathbf{z}}_0 = \mathbf{0}$. As can be seen in Fig. 12, in case of initializing with the denoised image, the algorithm converges faster. However, with simple zero initialization, we obtain the same result only after more CG iterations. The proposed algorithm is not sensitive to the initialization.

4) Effect of Oracle Scenario on the Performance of the Proposed Algorithm: It is instructive to show reconstructions starting from the oracle scenario where the true images are used for the weights and applied to noisy data. we consider a deblurring scenario in which test images are synthetically

TABLE V
PSNR ORACLE PERFORMANCE OF THE PROPOSED ALGORITHM VS. THE LOW-NOISE CASE

Image	Girl	Motocross bikes	Street	Building
Oracle	33.06	27.27	29.07	28.43
Algorithm 1	32.42	26.29	28.11	27.95

TABLE VI
ISNR VALUES FOR THE GIRL IMAGE AND OUT-OF-FOCUS BLUR WITH DIFFERENT NOISE STANDARD DEVIATION WHEN THE WEIGHTS ARE COMPUTED FROM THE INPUT NOISY BLURRED IMAGE

	$\sigma = 1$	$\sigma = 3$	$\sigma = 6$	$\sigma = 9$	$\sigma = 20$	$\sigma = 30$
ISNR	4.20	3.63	3.73	4.25	6.54	8.87

blurred with 25×25 Gaussian blur kernel with standard deviation 1.6 and the additive white Gaussian noise with variance 0.4 is added. Then, we compare the performance of the proposed algorithm, when the similarity weights are computed from the oracle images compared to the cases when these weights are derived from the given input (blurred, noisy) images. The results are summarized in Table V. As expected, using the oracle image for computing the similarity weights improves the performance of the algorithm. Also, it can be seen that for practical cases when the amount of noise is low, the results of the algorithm are not much different from their oracle counterparts.

5) Effect of the Noise Level in the Computation of the Similarity Weights: We add an experiment considering the effect of noise in computing the similarity weights. The Girl image is blurred by out-of-focus blur (of radius 7), and different amounts of noise (up to standard deviation equal to 30) are added to the blurred image. We fix all the parameters for different noise levels except the regularization parameter η which is changed proportional to the noise standard deviation as $h = 0.05\sigma$. In all the experiments, the similarity weights are computed from the noisy blurred input image in the first step of the algorithm. Table VI shows the ISNR values for different noise levels.[11] As can be seen even for very high noise levels and with computation of the weights from the input image, the algorithm is able to provide improvement with respect to the input degraded image. Although one can assume that the algorithm will fail at some noise level, we can conclude that it is robust enough to the errors in the input image for computing the similarity weights.

C. Synthetic Motion Blur

For assessing the performance of the algorithm in case of motion deblurring, we use the complex camera motion blur kernel provided by Shan et al. [22]. Again, noise with variances equal to 0.2 and 1 is added to the blurred images. In this case, we compare the proposed algorithm with two of the best available non-blind motion deblurring works. Periodic boundary conditions are used in these examples [1]. Also, we use patch size of 5×5, search neighborhood size of 11×11, number of outer iterations equal to 3, and the step decrement of the number of inner iterations (nDec) equal to 30 in these experiments. The values of the parameters η, β, h, and maximum number of inner CG iterations have been summarized in Table IV. The parameters of all algorithms are set for best performance. Table VII illustrates the quantitative results in this case. From the numerical results in Table VII, it is evident that our proposed kernel similarity-based algorithm shows very good performance in the case of non-blind nonlinear motion deblurring.

[11] ISNR is defined as the difference between the PSNRs of the output image and the input noisy blurred image.

TABLE VII
SSIM AND PSNR PERFORMANCE OF THE KERNEL SIMILARITY-BASED ALGORITHM IN COMPARISON WITH THOSE OF MOTION DEBLURRING METHODS IN [20] AND [22] FOR SYNTHETIC CAMERA MOTION BLUR. IN EACH CELL, THE FIRST NUMBER DENOTES SSIM VALUE, AND THE SECOND NUMBER REPRESENTS PSNR VALUE

Image	$\sigma^2 = 0.2$			$\sigma^2 = 1$		
	ours	[20]	[22]	ours	[20]	[22]
Building	0.9743 29.47	0.9734 29.74	0.9741 29.51	0.9525 28.45	0.9507 28.48	0.9572 28.45
Motocross Bikes	0.9756 28.04	0.9731 28.26	0.9692 26.95	0.9589 26.89	0.9470 26.45	0.9496 26.51
Girl	0.9636 33.54	0.9624 33.53	0.9586 33.15	0.9354 32.15	0.9351 31.99	0.9379 32.08
Street	0.9791 31.11	0.9766 31.50	0.9741 31.00	0.9584 29.75	0.9507 29.32	0.9558 29.35
Boat	0.9634 29.91	0.9687 29.81	0.9532 28.37	0.9342 28.57	0.9311 28.38	0.9344 28.27
Book shelf	0.9857 28.81	0.9839 29.24	0.9821 28.42	0.9711 27.31	0.9680 27.61	0.9693 27.21

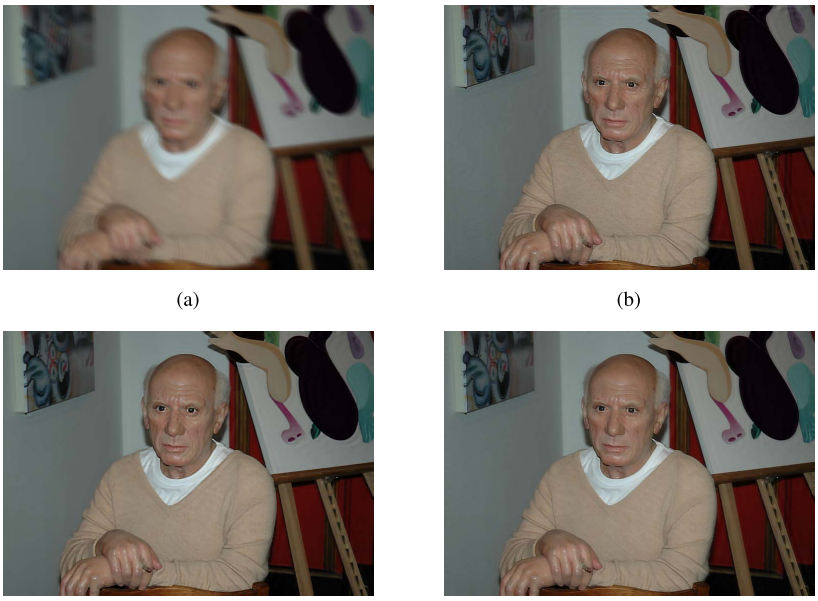
D. Real Motion Deblurring

Now, we deal with more challenging motion blur situations where the blur kernel is estimated using two of the existing blur kernel estimation methods [22], [48] from real motion blurred test images. The estimated blur kernels are used independently to derive the final deblurred images. The performance of the proposed method is compared with those of [20] and [22], which are among the best non-blind motion deblurring methods.

For all examples, the patch size and search window size are selected to be 5×5 and 11×11, respectively. Also, we use reflective boundary conditions for these experiments [1], [49]. Figures 13 and 14 show the outputs of different methods, when the blur kernels are estimated using the algorithm in [22]. Also, Figures 15 and 16 illustrate the results of different algorithms when applied to real blurred images using corresponding estimated blur kernels from [48]. As can be seen, our iterative deblurring algorithm produces high quality outputs as good as the state-of-the art.

E. Comparison to Traditional Normalized Graph Laplacian Through Image Deblurring Experiments

In order to compare the performance of our proposed algorithm with the traditional normalized Laplacian ($I - D^{-1/2} K D^{-1/2}$) formulation, we consider the following deblurring experiments. In these experiments, we first

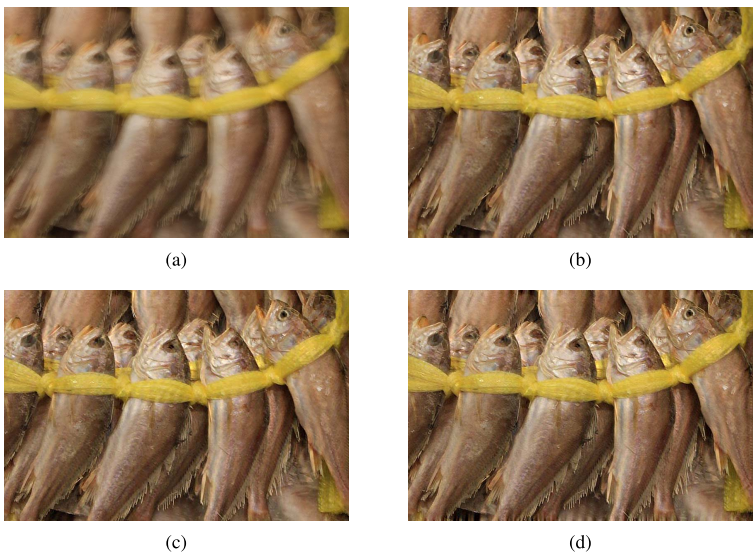


(a) (b) (c) (d)

Fig. 13. Real motion deblurring example: (a) input blurred noisy image, (b) Output of hyper-Laplacian algorithm [20], (c) output of [22], and (d) output of our algorithm ($\eta = 0.031$, $\beta = 0.6$).

(a) (b) (c) (d)

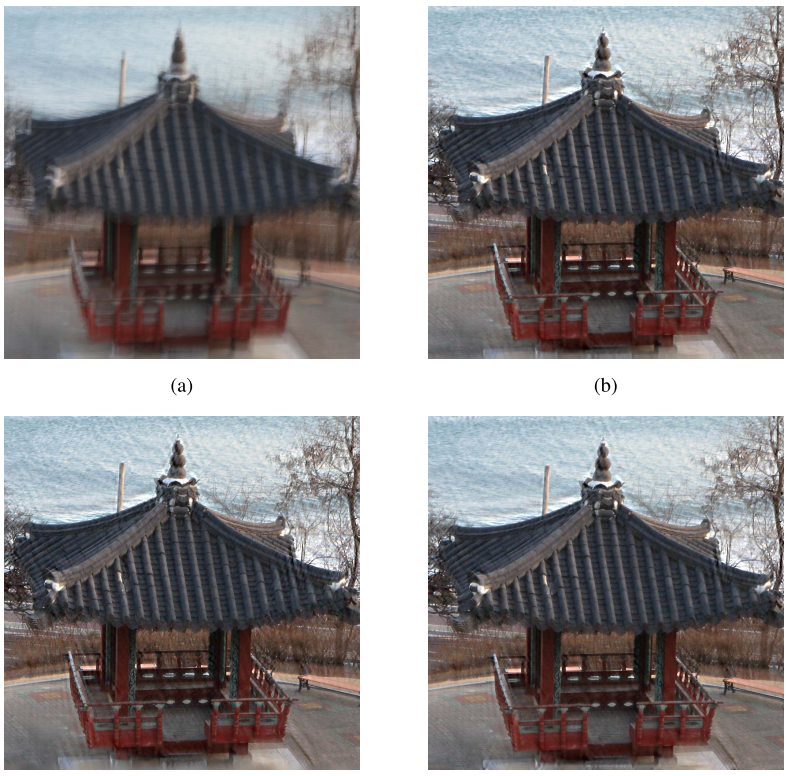
Fig. 14. Real motion deblurring example: (a) input blurred noisy image, (b) Output of hyper-Laplacian algorithm [20], (c) output of [22], and (d) output of our algorithm ($\eta = 0.018$, $\beta = 0.9$).



(a) (b) (c) (d)

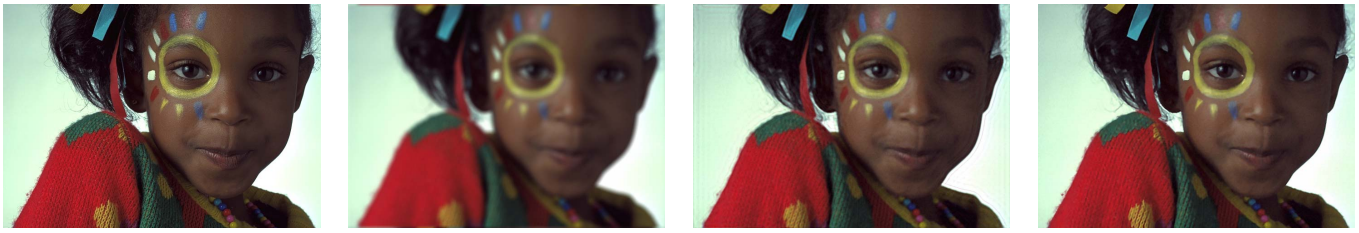
Fig. 15. Real motion deblurring example: (a) input blurred noisy image, (b) Output of hyper-Laplacian algorithm [20], (c) output of [22], and (d) output of our algorithm ($\eta = 0.031$, $\beta = 0.6$).

convolve the Girl image with Gaussian blur (standard deviation equal to 1.6) and add white Gaussian noise with standard deviation $\sigma = 1$ to it. Then, we use the proposed deblurring algorithm to get the desired estimate. Also, we consider a similar cost function to the one in (16) using the traditional normalized Laplacian $I - W_D = I - D^{-1/2} K D^{-1/2}$



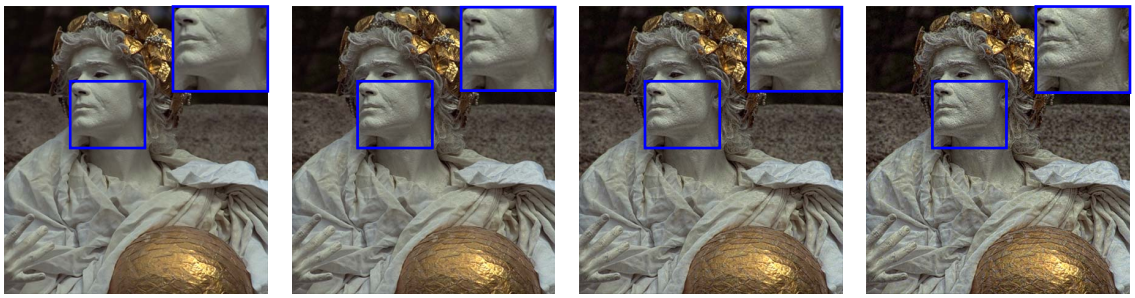
(a) (b) (c) (d)

Fig. 16. Real motion deblurring example: (a) input blurred noisy image, (b) Output of hyper-Laplacian algorithm [20], (c) output of [22], and (d) output of our algorithm ($\eta = 0.032$, $\beta = 0.6$).



(a) (b) (c) (d)

Fig. 17. Deblurring examples with blurred noisy Girl image by out-of-focus blur with radius 7 and additive white Gaussian noise with standard deviation $\sigma = 1$: (a) clean image, (b) blurred noisy image, (c) output of the deblurring algorithm with the corresponding traditional normalized Laplacian (PSNR = 29.40dB, SSIM = 0.8734), and (d) output of our proposed deblurring algorithm (PSNR = 30.58dB, SSIM = 0.9058).



(a) (b) (c) (d)

Fig. 18. Image sharpening example for a synthetic image with moderate Gaussian blur; (a) input blurred image, (b) output of (24) with $\beta = 0.7$, (c) $\beta = 1.3$, and (d) $\beta = 2.3$.

as:

$$E_D(\mathbf{z}) = (\mathbf{y} - A\mathbf{z})^T\{I + \beta(I - W_D)\}(\mathbf{y} - A\mathbf{z}) + \eta\, \mathbf{z}^T(I - W_D)\mathbf{z}, \quad (26)$$

in which the normalized matrix $W = C^{-1/2}KC^{-1/2}$ has been replaced by $W_D = D^{-1/2}KD^{-1/2}$. The same set of parameters are used for both cases. The results are shown in Fig. 17. As can be seen, our proposed normalized graph Laplacian produces deblurring output with higher quality, verifying the

(a) (b)

Fig. 19. Sharpening example for a real image; (a) input image and (b) sharpened image using our algorithm ($\beta = 1.3$).

superiority of the proposed normalized graph Laplacian with respect to its traditional counterpart.

F. Image Sharpening Examples

There are many circumstances in which the image has undergone moderate amount of blurring and there is no explicit knowledge of the blurring kernel available. In such cases, it is possible to exploit the data-adaptive sharpening framework in Section V to produce a sharper and more pleasant output from the slightly degraded input image. In this case, the sharpened images derived through our kernel similarity-based method have been shown in Fig. 18 for synthetic test images and different values of the parameter β. As can be seen in Fig. 18, as β increases, the high-pass filtering property of $I + \beta(I - W)$ strengthens, and sharper images are produced. This experiment demonstrates the effect of the parameter β in the filtering term $I+\beta(I-W)$ in our general framework, such that for $\beta > 0$, the behavior of $I + \beta(I - W)$ tends towards a high pass filter. Also, Fig. 19 illustrates the output of our sharpening method when applied to a real test image.

VIII. Conclusion

In this paper, we proposed a broad framework for kernel similarity-based image restoration. We have introduced a new objective function for image enhancement by coupling the data and prior terms via structurally encoded filtering and Laplacian matrices. Also, we have presented a graph-based filtering interpretation of the proposed method, providing better intuition for data-adaptive approaches as well as a path for further improvement of such approaches. Through experiments, the effectiveness of the kernel similarity-based method has been verified for a range of blurring scenarios via comparison with some of the existing state-of-the-art algorithms. Also, special cases within the proposed framework were highlighted for image sharpening and denoising. This kernel similarity-based approach is general enough to be exploited for many different restoration tasks, as long as there is a reasonable way to estimate the kernel similarity matrix K.

For future works, this approach can be extended to blind image deblurring, where the kernel similarity framework is applicable for estimating the underlying blur kernel as well. Also, this data-adaptive work can be applied to more complicated non-uniform blur situations. Preconditioned CG can also be used to further improve the convergence properties of the proposed algorithm.

References

[1] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, 1st ed. Philadelphia, PA, USA: SIAM, 2006.

[2] M. Donatelli, C. Estatico, A. Martinelli, and S. Serra-Capizzano, "Improved image deblurring with anti-reflective boundary conditions and re-blurring," *Inverse Problems*, vol. 22, no. 6, p. 2035, 2006.

[3] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, Jun. 2007.

[4] M. Zibulevsky and M. Elad, "L1–L2 optimization in signal and image processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 76–88, May 2010.

[5] G. Chantas, N. P. Galatsanos, R. Molina, and A. K. Katsaggelos, "Variational Bayesian image restoration with a product of spatially weighted total variation image priors," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 351–362, Feb. 2010.

[6] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248–272, Aug. 2008.

[7] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Adaptive total variation image deblurring: A majorization–minimization approach," *Signal Process.*, vol. 89, no. 9, pp. 1683–1693, 2009.

[8] G. Peyré, S. Bougleux, and L. Cohen, "Non-local regularization of inverse problems," in *Proc. 10th Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2008, pp. 57–68.

[9] X. Zhang, M. Burger, X. Bresson, and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *SIAM J. Imag. Sci.*, vol. 3, no. 3, pp. 253–276, 2010.

[10] H. Takeda, S. Farsiu, and P. Milanfar, "Deblurring using regularized locally adaptive kernel regression," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 550–563, Apr. 2008.

[11] R. N. Neelamani, H. Choi, and R. Baraniuk, "ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 418–433, Feb. 2004.

[12] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[13] S. Lefkimmiatis, A. Bourquard, and M. Unser, "Hessian-based norm regularization for image restoration with biomedical applications," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 983–995, Mar. 2012.

[14] J. Ni, P. Turaga, V. M. Patel, and R. Chellappa, "Example-driven manifold priors for image deconvolution," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3086–3096, Nov. 2011.

[15] T. S. Cho, C. L. Zitnick, N. Joshi, S. B. Kang, R. Szeliski, and W. T. Freeman, "Image restoration by matching gradient distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 683–694, Apr. 2012.

[16] 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.

[17] X. Li, "Fine-granularity and spatially-adaptive regularization for projection-based image deblurring," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 971–983, Apr. 2011.

[18] A. Danielyan, V. Katkovnik, and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, Apr. 2012.

[19] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007, Art. ID 70.

[20] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Advances in Neural Information Processing Systems 22*. Red Hook, NY, USA: Curran Associates, 2009, pp. 1033–1041.

[21] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Progressive inter-scale and intra-scale non-blind image deconvolution," *ACM Trans. Graph.*, vol. 27, no. 3, Aug. 2008, Art. ID 74.

[22] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–10, Aug. 2008, Art. ID 73.

[23] S. Bougleux, A. Elmoataz, and M. Melkemi, "Local and nonlocal discrete regularization on weighted graphs for image and mesh processing," *Int. J. Comput. Vis.*, vol. 84, no. 2, pp. 220–236, 2009.
[24] J. Boulanger, C. Kervrann, P. Bouthemy, P. Elbau, J.-B. Sibarita, and J. Salamero, "Patch-based nonlocal functional for denoising fluorescence microscopy image sequences," *IEEE Trans. Med. Imag.*, vol. 29, no. 2, pp. 442–454, Feb. 2010.
[25] G. Gilboa and S. Osher, "Nonlocal linear image regularization and supervised segmentation," *Multiscale Model. Simul.*, vol. 6, no. 2, pp. 595–630, 2007.
[26] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, 2008.
[27] F. G. Meyer and X. Shen, "Perturbation of the eigenvectors of the graph laplacian: Application to image denoising," *Appl. Comput. Harmon. Anal.*, vol. 36, no. 2, pp. 326–334, 2014.
[28] A. D. Szlam, M. Maggioni, and R. R. Coifman, "Regularization on graphs with function-adapted diffusion processes," *J. Mach. Learn. Res.*, vol. 9, pp. 1711–1739, Jun. 2008.
[29] G. Taubin, "A signal processing approach to fair surface design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn.*, 1995, pp. 351–358.
[30] P. A. Knight and D. Ruiz, "A fast algorithm for matrix balancing," *IMA J. Numer. Anal.*, vol. 33, pp. 1029–1047, 2013.
[31] A. Kheradmand and P. Milanfar, "A general framework for kernel similarity-based image denoising," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2013, pp. 415–418.
[32] P. Milanfar, "Symmetrizing smoothing filters," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 263–284, 2013.
[33] P. Knopp and R. Sinkhorn, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific J. Math.*, vol. 21, no. 2, pp. 343–348, 1967.
[34] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
[35] F. R. Chung, *Spectral Graph Theory*, vol. 92. Providence, RI, USA: AMS, 1997.
[36] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[37] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1047–1060, Jul. 2008.
[38] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[39] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Statist.*, vol. 36, no. 3, pp. 1171–1220, 2008.
[40] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.
[41] 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.
[42] M. Meila and J. Shi, "A random walks view of spectral segmentation," in *Proc. 8th Int. Workshop Artif. Intell. Statist. (AISTATS)*, 2001, pp. 1–6.
[43] J. Immerkær, "Fast noise variance estimation," *Comput. Vis. Image Understand.*, vol. 64, no. 2, pp. 300–302, 1996.
[44] P. C. Hansen, "Regularization tools version 4.0 for Matlab 7.3," *Numer. Algorithms*, vol. 46, no. 2, pp. 189–194, 2007.
[45] 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.
[46] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterington, "A study of methods of choosing the smoothing parameter in image restoration by regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 326–339, Apr. 1991.
[47] M. Hussein, F. Porikli, and L. Davis, "Kernel integral images: A framework for fast non-uniform filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[48] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, Dec. 2009, Art. ID 145.
[49] J. G. Nagy, K. Palmer, and L. Perrone, "Iterative methods for image deblurring: A MATLAB object-oriented approach," *Numer. Algorithms*, vol. 36, no. 1, pp. 73–93, 2004.
[50] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.
[51] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 171–184.

Amin Kheradmand (S'12) received the B.S. degree in electrical engineering from the Isfahan University of Technology, Isfahan, Iran, in 2008, and the M.S. degree in electrical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2011. He is currently pursuing the Ph.D. degree in electrical engineering with the University of California at Santa Cruz, Santa Cruz, CA, USA.

His research interests are in signal, image and video processing, and computational photography (deblurring, denoising, and superresolution).

Peyman Milanfar (F'10) received the bachelor's degree in electrical engineering and mathematics from the University of California at Berkeley, Berkeley, CA, USA, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA. Until 1999, he was with SRI International, Menlo Park, CA, USA, and was a Consulting Professor of Computer Science with Stanford University, Stanford, CA, USA. He has been with the Faculty of Electrical Engineering, University of California at Santa Cruz, Santa Cruz, CA, USA, since 1999, and served as an Associate Dean of the School of Engineering from 2010 to 2012. Since 2012, he has been on leave with the Google Research and Google-X, Mountain View, CA, USA, where he was involved in computational photography, and in particular, the imaging pipeline for the Google Glass. He was a recipient of the National Science Foundation CAREER Award in 2000 and the Best Paper Award from the IEEE Signal Processing Society in 2010.